Similarity and Diversity in Information Retrieval

by

John Akinlabi Akinyemi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

John Akinlabi Akinyemi

Abstract

Inter-document similarity is used for clustering, classification, and other purposes within information retrieval. In this thesis, we investigate several aspects of document similarity. In particular, we investigate the quality of several measures of inter-document similarity, providing a framework suitable for measuring and comparing the effectiveness of inter-document similarity measures. We also explore areas of research related to novelty and diversity in information retrieval. The goal of diversity and novelty is to be able to satisfy as many users as possible while simultaneously minimizing or eliminating duplicate and redundant information from search results. In order to evaluate the effectiveness of diversity-aware retrieval functions, user query logs and other information captured from user interactions with commercial search engines are mined and analyzed in order to uncover various informational aspects underlying queries, which are known as subtopics. We investigate the suitability of implicit associations between document content as an alternative to subtopic mining. We also explore subtopic mining from document anchor text and anchor links. In addition, we investigate the suitability of inter-document similarity as a measure for diversity-aware retrieval models, with the aim of using measured inter-document similarity as a replacement for diversity-aware evaluation models that rely on subtopic mining. Finally, we investigate the suitability and application of document similarity for requirements traceability. We present a fast algorithm that uncovers associations between various versions of frequently edited documents, even in the face of substantial changes.

Supervisor:
Charles Clarke

Examiners:
Mark Smucker, Daniel Berry, Olga Vechtomova, and Vlado Keselj

Acknowledgements

First and foremost, I would like to thank my supervisor, Charles Clarke, for providing such an exceptional research collaboration environment during my doctoral programme at the University of Waterloo. Charlie provided over and beyond the financial support that is expected when my major funding finished. He is a vibrant, brilliant, and consistently motivated advisor providing such an excellent and insightful academic advice and career guidance.

I would like to thank the other members of my thesis examination committee — Daniel Berry, Mark Smucker, and Olga Vechtomova — for the sacrifice of their time reading my thesis, providing insightful feedback and attending my thesis defence. I also want to thank Vlado Keselj for traveling all the way from Nova Scotia to attend my thesis defence and serve as my external examiner.

To all my colleagues at the IR/PLG Lab, I say thank you for making it a pleasant place. Especially, I would like to thank Ashif Harji, Maheedhar Kolla, Mona Mojdeh, Brad Lushman, and Azin Ashkan.

A big thank you goes to my family and friends. They consistently helped me remember and focus on what is important in life. In particular, I would like to thank my wife, Elizabeth Olabisi, for everything you gave and sacrificed to support me in the graduate school. I would also like to thank Akanni Akinyemi, Segun and Rachael Aminu, Tunde and Funmi Akindipe, Femi and Favour Olumofin, Timi Olaatanda, Mary Eluobaju, and my mother-in-law, Christanah Famuyide, for her support. I am grateful to Steve and Beth Fleming and the entire body of Koinonia Christian Fellowship, for providing a safe place for worship, fellowship, friendship, service, and growth.

Finally, I also acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a NSERC Postgraduate Scholarship, the University of Waterloo through a President's Graduate Scholarship, and other financial sources provided through my supervisor.

Dedication

I dedicate this dissertation to my wonderful family — *Elizabeth Olabisi*, *Toluwanimi Shalom*, and *Sara-Anne Ayobami*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

1.1.1 Document Similarity

Document similarity is used extensively in information retrieval for various purposes (Smucker and Allan, 2006; Kurland and Lee, 2004; Meister et al., 2010; Akinyemi and Clarke, 2011; Cooper et al., 2002; Lin, 2009; Uzuner et al., 2004). The similarity between two objects may be inferred from what they have in common as well as the differences between them (Tversky, 1977; Lin, 1998; Aslam and Frost, 2003). The more commonality the objects share, the more similar they are, and the more differences between them, the less similar they are. In this thesis, we explore similarity measures in general, and specifically effectiveness measures for document similarity.

In a typical *adhoc* search operation, when a search user represents her information need with a query, a search engine returns a list of documents from the corpus ranked according to their relevance to the user's query. The relevance of a document to a given query describes a notion of their relatedness, association, and similarity with the query. Despite the importance of document relevance, a search user would also prefer that the search engine minimizes information redundancy within the retrieved result. Ideally, each document on the list should provide additional information different from the information provided by documents at higher ranks that

would have already been seen by the user. The problem of reducing information redundancy and providing multi-faceted result in search, which is referred to as *diversity* and *novelty* has been identified and is being actively investigated in information retrieval research. We examine some ideas in diversity and novelty.

An interesting area also examined concerns frequently edited documents, such as versions of source code artifacts and wiki applications, i.e. applications that store versions of frequently edited documents. Within a wiki or source code document, a user might need to know how a certain arbitrary portion of the document has changed over time. The user may require the change information that has occurred over time between two versions of a document at an arbitrary document location. This concept is referred to as *traceability* in software maintenance and *data provenance* in databases. We examine this area of research in more details within the context of document similarity.

1.1.2 Applications of Document Similarity

Clustering, classification, and generalization are application areas of document similarity. Documents are grouped together because they are either related or they satisfy same information request (query). Document similarity is used for ranking in information retrieval (Ponte and Croft, 1998; Kurland, 2006). Smucker and Allan (2006) expressed document similarity as a "find similar" problem. Lin (2009) calculated pairwise document similarity in a very large dataset using MapReduce (Dean and Ghemawat, 2004) in order to investigate the "more like this" problem in PubMed dataset. Kurland and Lee (2004) and Meister et al. (2010) employed inter-document similarity within clusters of similar documents as an additional feature for improving the effectiveness of an adhoc retrieval function. Akinyemi and Clarke (2011) utilized document similarity for performing evaluation for diversity-aware retrieval functions.

Duplicate document detection algorithms rely on document similarity (Cooper et al., 2002). The extracted shingles (Broder et al., 1997) constitutes the feature for estimating a measure of association between document pairs. Hashing (Stein, 2007; Stein and Potthast, 2007) and fingerprinting (Heintze, 1996) also rely on a notion of similarity between hashed values and fingerprints of text fragments in documents. In software engineering and maintenance, document similarity is

used for code clone detection (Duala-Ekoko and Robillard, 2007), as well as recovering and tracing association links between software artifacts (Hayes et al., 2007; Cleland-Huang et al., 2009; Antoniol et al., 2002). Other researchers have used document similarity for detecting similarity of document contents (Uzuner et al., 2004) such as plagiarism and copyright infringement detection. Document similarity is also very important for document deduplication (Bhattacharya and Getoor, 2004), in which case duplicate and near-duplicate documents are avoided. In addition, document similarity is used for co-reference resolution (Elsayed et al., 2008) in text corpora.

1.1.3 Similarity Measures

Information needs are generally represented as queries (Baeza-Yates, 2005) and provided to information retrieval (IR) systems (i.e. search engines). An IR system is required to sieve through all the documents in the corpus and provide a ranked list of the most relevant documents to the query. This process is referred to as adhoc retrieval. It is important to evaluate the effectiveness of various retrieval models and be able to compare their performances. The IR community has embraced the Cranfield approach for evaluating retrieval and ranking models (Manning et al., 2008; Harter and Hert, 1997; Büttcher et al., 2010) by which the top k retrieved documents from different retrieval systems are pooled and randomly presented to human editorial assessors for relevance judgment. Documents judged relevant by editorial assessors are accepted as relevant documents for the query. Apart from the concept of similarity in adhoc retrieval tasks where relevant documents are retrieved for queries, another important concept of similarity in text processing is inter-document similarity, which is also referred to as pairwise document similarity. Inter-document similarity measures the similarity between two documents. No query is specified in inter-document similarity, and document ranking is not required.

Central to document similarity computation is the set of features that are implicitly encoded in documents. Features are the characteristics and properties of documents such as the frequency of terms, proximity between terms, term synonymy, document length, topic of the document, and other document and corpus characteristics. In order to obtain the similarity between document pairs, their features are utilized as a means of comparison. Functions that perform actual document–document comparison are referred to as document similarity measures. Document ranking functions exploit document characteristics and features. Document similarity is the ba-

sis behind ranking functions in which case documents having more topical (or other features) similarity or relevance are retrieved for a given query (Büttcher et al., 2010; Croft et al., 2009).

The foundation of document similarity in information retrieval is the Cluster Hypothesis (van Rijsbergen, 1979) which states that *closely associated documents tend to be relevant to the same requests*. Van Rijsbergen (1979) claims documents are grouped together because they are related to each other and because they satisfy the same information request. Retrieval and ranking functions depend on mathematical, probabilistic, statistical and generative language, machine learning and empirical models. A commonality between these retrieval and ranking models is their suitability for adhoc retrieval in which case queries are short compared to the length of documents in the corpus. A notable difference between the generative language model and other retrieval models is the concept of smoothing. Smoothing attempts to eliminate or reduce the data sparsity problem which results because short queries have a lot more zero term frequencies than term frequencies in long documents. For short queries, smoothing assigns very small frequency values to document terms that are not present in the queries such that zero probabilities are avoided. We refer to this category of short-queries-and-long-documents retrieval as query–documents (Q–D) retrieval problem. Apart from Q–D retrieval tasks, there are situations when document pairs are compared for similarity. Computing the similarity between document pairs, i.e. document–document (D–D) similarity is referred to as pairwise document similarity (PDS) or inter-document similarity.

There are standard measures for evaluating the effectiveness and quality of Q–D similarity functions (i.e. retrieval functions). Some of these measures include Precision, Recall, Average Precision (AP), Mean Average Precision (MAP), Discounted Cumulative Gain (DCG), normalized DCG (nDCG), F-measures, Subtopic Recall, Mean Reciprocal Rank (MRR), and others. Evaluating the effectiveness of D–D similarity functions have received a relatively less attention. In fact information retrieval evaluation frameworks such as TREC[1] have paid less attention to the performance evaluation of D–D similarity functions than Q–D similarity evaluation functions. Regardless, there are some research efforts towards the creation of evaluation methods for D–D similarity (Voorhees, 1985; Smucker and Allan, 2009; Aslam and Frost, 2003). These methods rely on results obtained from Q–D adhoc retrieval tasks as their evaluation models attempt to recast the problem of evaluating the performance of inter-document similarity as a form

[1] `trec.nist.gov`

of Q–D retrieval task by making some assumptions about the relevance between (i) relevant and relevant documents at higher top k ranks, and (ii) relevant and non-relevant documents. The work in this thesis is predominantly related to D–D similarity functions.

1.1.4 Diversity

In diversity and novelty-aware ranking models, retrieved results should satisfy the diversity in user needs as well as reduce redundancy in retrieved documents. Queries that have multiple interpretations are called ambiguous queries. For example the query *windows* is ambiguous because it could mean a glass window, the Microsoft Windows operating systems, replacement windows, X Windows, Windows musical group, or the Windows movie. Under-specified queries on the other hand may be an unambiguous query having various aspects. For example the query *Microsoft Windows* is unambiguous but there are various aspects to the query such as the Microsoft company, Microsoft Windows operating system, versions of Microsoft Windows operating system, Microsoft Windows application updates, and other concepts related to Microsoft Windows.

When a query is ambiguous, result diversification is necessary (Clarke et al., 2008) in order to cater for the variety of all search users that use the same query but have different information needs. In the *windows* query example, some users might be satisfied with information about Microsoft Windows operating systems, others might be interested in only replacement windows, X Windows, Windows musical group, or the Windows movie. Work in diversity attempts to maximize the utility of search results in order to satisfy as many user needs as possible. When a query is under-specified, the result should cover various aspects of the query or contain novel information about the same aspect of the query. The result from the *Microsoft Windows* query should contain information about various aspects of Microsoft Windows. Apart from satisfying the information need of diverse users, it is equally important that each user group obtains fresh, new, and relevant information at each rank in the top k retrieved documents. This aspect of information freshness is referred to as *novelty* (Clarke et al., 2008).

Within the context of novelty and diversity, the cluster hypothesis might be reformulated to cater for information requests having several interpretations and various aspects. This diversity-oriented cluster hypothesis might be stated as "*closely associated documents tend to be relevant*

to the same interpretations and aspects of an ambiguous and under-specified information request". Research in diversity and novelty basically addresses problems in diversity-aware ranking models as well as evaluation models suitable for measuring the effectiveness of diversity-aware ranking models. The thesis provides a diversity-aware re-ranking model as well as an evaluation technique that is modeled on inter-document similarity between relevant documents.

1.1.5 Traceability

Over time, frequently edited documents such as source code and Wikipedia evolve to contain text and information not originally present in them. Document evolution occurs when there have been significant edits such as text insertions, deletions, and when text fragments are moved around within the same document and between different documents. Given source code that implements a particular software requirement, a software engineer may wish to establish a link from the location in a text document where the requirement is specified to the location in a source code file where the requirement is implemented. Traceability is the ability to discover and trace the source or origin of text fragments from a given document location. Apart from tracing origins of text fragments, traceability also includes the discovery and tracing of other documents and text fragments having significant and non-trivial associations with the text in the given location. For example, if two consecutive change-causing edits occur to a Wikipedia article, such that the title is changed between Wikipedia versions and a paragraph in the article (we refer to the paragraph as an edit location) is also copied to another article. On a more recent Wikipedia version, a trace of this edit location should be able to locate the article having the new title as well as the other article sharing the same paragraph.

We investigate the suitability of document similarity approaches for discovering and recovering traceable links among versions of frequently edited documents. We provide more details on this subject in Chapter 5.

1.1.6 Evaluation Frameworks

In this section, we provide an overview of frameworks used for evaluating the effectiveness of ranking functions used in this thesis. We also provide an overview of some of the diversity-aware

evaluation measures reported in subsequent results from our experiments.

TREC

TREC[2] is an acronym for Text REtrieval Conference. The TREC conference began in 1992 and has since been sponsored on a yearly basis by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense. TREC is a collaborative framework that supports research in information retrieval. It provides an infrastructure for evaluating information retrieval models.

NIST provides test collections consisting of corpus and topics for TREC experimental tasks. In addition, specific problems in information retrieval are modeled into tasks, topics are designed for each task, and participants are encouraged to develop retrieval models to address each of the tasks using the data sets and topics provided by TREC organizers. Participants in TREC tracks submit the result from their retrieval system to NIST for effectiveness judgment. The collection, submitted results and judgments facilitate repeatability of experimental methods as well as a standard comparative measurement framework for the information research community.

NTCIR

NTCIR[3] is the acronym for National Institute of Informatics (NII) Test Collection for Information Retrieval Systems. The NTCIR Workshop which commenced in 1997 provides another evaluation framework for research in Information Access (IA) technologies including information retrieval, question answering, text summarization and information extraction. NTCIR is sponsored by the Japan Society for Promotion of Science (JSPS), Research Center for Information Resources at the National Institute of Informatics (RCIR/NII) and several other organizations at different times in the past. NTCIR Workshop tasks usually make use of documents written in Asian languages such as Japanese and Chinese. There are other evaluation frameworks such as CLEF and INEX, but we did not make use of them in this thesis.

[2] `http://trec.nist.gov/`

[3] `http://research.nii.ac.jp/ntcir/index-en.html`

Evaluation Measures

Tasks within these evaluation frameworks are judged for their effectiveness using the state-of-the-art evaluation measures. These are standardized evaluation measures within the information retrieval community. Some of the measures that may be used to evaluate diversity and novelty-aware ranking functions include α-nDCG, ERR-IA, and strec.

α-ndcg

One of the official measures used for evaluating result diversification in the TREC Web Track is α-nDCG. The α-nDCG measure makes use of the intents of an ambiguous or under-specified query — referred to as *nuggets* — to assign gain values to documents based on the nuggets they are relevant to. If a document is relevant to nuggets already seen at higher ranks (i.e., nuggets relevant to documents at higher ranks), they are penalized. The rank of a document is also used to discount the gain value.

ERR-IA

The ERR-IA evaluates a document as being dependent on every other relevant documents ranked higher than the current document. The information contribution provided by a document is compared with the information already provided by all other relevant documents at higher ranks. Actual document rank is also used as a discounting factor. As in α-nDCG, intents of an ambiguous or under-specified queries are also directly modeled into ERR-IA.

strec

The *strec* measure computes the subtopic recall value for each retrieval function. It is the number of distinct subtopics covered in the top *k* result of relevant documents. *strec* also relies on the availability of diverse subtopics of ambiguous and under-specified queries.

1.2 Open Problems

In this section, we describe some open problems identified in the areas of document similarity, diversity-aware retrieval and evaluation models, and traceability among frequently edited documents.

1.2.1 Document Similarity

Most research in document similarity either focus on query–document ranking models or evaluation models for query–document retrieval functions. Most work that incorporate inter-document similarity make use of the cosine similarity as a measure of inter-document similarity. Apart from the cosine similarity measure, there are other measures of similarity that may be suitable for computing inter-document similarity. These include the frequency-based approach such as the Dice coefficient and Jaccard index, language model-based approach, data compression-based approach, and information-theoretic approach.

In recent times, language models are being used for computing inter-document similarity. For example, Kurland (2006), Kurland and Lee (2004), and Meister et al. (2010) employed inter-document similarity within clusters of similar documents as an additional feature for improving the effectiveness of an adhoc retrieval function. The cosine similarity measure was used as query–document similarity function in the SMART (Salton and McGill, 1986) information retrieval system at the earlier stage of information retrieval. More sophisticated and better performing ranking functions have since replaced cosine similarity as a query–document ranking function. Unfortunately, the computation of document–document similarity to a large extent still relies on cosine similarity method even though there is no proof to support that cosine similarity is the best performing inter-document similarity measure.

In this thesis, we attempt to measure and compare the quality of various inter-document similarity measures using the subtopic overlap between relevant documents as a crude measure of inter-document similarity. Our goal is to first compare the effectiveness of various inter-document similarity measures and explore the creation of performance evaluation measures for inter-document similarity functions. We are of the opinion that the information retrieval research

community should provide a standard performance evaluation framework to measure the quality of inter-document similarity measures in the existing evaluation framework, such as TREC.

1.2.2 Diversity

As previously mentioned, the introduction of diversity into information retrieval models attempts to cater for the various interpretations of ambiguous queries. Likewise, it is essential that retrieval results cover various non-trivial aspects of under-specified queries. In order to evaluate the performance of diversity-aware ranking functions, various interpretations and aspects of ambiguous and under-specified queries are obtained and explicitly included in models that evaluate the effectiveness of diversity-aware ranking functions. These interpretations and aspects are referred to as *subtopics*. Subtopics are provided by the organizers of diversity-oriented retrieval tasks as part of the evaluation frameworks in both TREC and NTCIR.

All current diversity-aware evaluation measures such as α-ndcg (Clarke et al., 2008), Precision-IA (Agrawal et al., 2009), ERR-IA (Chapelle et al., 2009) and $D\#$ (Sakai and Song, 2011) require explicit subtopic categorization of relevant documents. For ambiguous queries, subtopic definition is a necessary requirement. Therefore, subtopics are defined for queries.

Evaluation models rely on subtopics in order to be effective. There are two fundamental problems with this approach. First, subtopics (query intents) are obtained from user interaction data mined from the query and click logs of commercial search engines (Radlinski et al., 2010). Unfortunately, query and click logs are not generally available for the research community, and when available, usage is limited and restricted. This makes the subtopic mining process non-repeatable and dependent on the particular search engine in consideration.

Secondly, categorizing relevant documents into subtopics is very expensive to do because human expert editorial judges are saddled with the responsibility of assigning subtopics to documents. Relevant documents submitted by participating research groups in the evaluation framework runs into tens of thousands. Judging this huge amount of documents correctly and timely is a challenging endeavor which is also expensive. Apart from the high cost of obtaining subtopic categories, human involvement in the process might introduce human errors. We also note that not all relevant documents are judged and there is no guarantee that all documents are judged correctly.

On the first problem, there is need to explore alternative approaches that does not rely solely on query and click logs of commercial search engines. An alternative approach should strive to obtain all its input directly from the available dataset without depending on the data provided by commercial search engines. It is important that the approach is repeatable and based on generally available resources.

This thesis investigates alternative approaches for mining subtopics and discovering query intents directly from text corpora. Concerning the second problem whereby diversity-aware evaluation measures depend on explicit subtopic categorization, there is need to explore alternative evaluation methods that do not rely on subtopic categorization. These evaluation measures should limit or completely avoid errors that may be introduced because of human involvement. To a large extent, categorizing documents into subtopics is a subjective exercise. This is even demonstrated through the non-agreement of judgments performed by different editorial assessors. Obviously, objectivity is required in results of scientific experiments. This is another problem area addressed by the work in this thesis.

1.2.3 Traceability

Maintaining a link for a specific edit location between a source and target documents may be achieved by annotating the location of an edit on the target with a unique tag such that the unique tag becomes the link. This type of links that are manually generated and maintained are referred to as hard links. Maintaining hard links can quickly become burdensome on the maintainers as a result of several changes especially when portions of documents are moved around, when source code is refactored, or when there is article update and clean-ups in Wikipedia. We identified link maintenance as an important problem as documents evolve over time. This thesis investigates an automatic approach for maintaining links between edit locations in a source document and the corresponding locations in their target documents.

1.3 Contributions

This thesis explores document similarity and diversity in general. Our contributions cut across models of evaluation for measures of inter-document similarity and diversity-aware retrieval and ranking models, corpus-based intent discovery methods, diversity-aware retrieval methods, and robust link discovery and maintenance for tracing frequently edited documents. We investigate evaluation measures for inter-document similarity, intent discovery, diversity-aware retrieval models, and subtopic categorization. A summary of the main contributions of this thesis is presented as follows:

1.3.1 Inter-document similarity measures

We made two contributions to inter-document similarity measures. First, with respect to subtopic categorization of relevant documents done for diversity-aware ranking models, we investigate the quality of subtopic judgment categorization performed by human editorial assessors. We investigated whether there is positive correlation between subtopic judgment and measured inter-document similarity. If documents sharing common subtopics also have higher inter-document similarity scores than those sharing no common subtopics, one might be able to draw a relationship between subtopic overlap among relevant documents and their measured similarity. Our second contribution attempts to provide a framework for evaluating the effectiveness of various inter-document similarity measures.

Do subtopic judgments correlate with inter-document similarity?

We investigate the correlation between inter-document similarity measures and subtopic categorization of relevant documents carried out by human editorial judges. Ideally, there should be a positive correlation between measured, inter-document similarity and relevance judgments by human editorial assessors. The correlation should be a reflection of the cluster hypothesis, i.e. documents sharing common subtopics should have higher inter-document similarity values than those having no common subtopic. We investigate this idea experimentally using data and test collection provided by TREC. Result from our experiment showed a positive correlation between

values of measured inter-document similarity and the frequency of subtopics that are commonly shared by relevant documents. The more subtopics shared by documents, the more similar they are. Therefore, we conclude that the quality of subtopic categorization performed by human assessors is mostly satisfactory.

Evaluating inter-document similarity measures

Since subtopic judgments correlate with measured inter-document similarity to a reasonable extent, we accept it as a crude similarity measure which inter-document similarity functions may be evaluated against. We enlisted several inter-document similarity measures to perform inter-document similarity for the set of relevant documents in both the 2009 and 2010 diversity task of the TREC Web Track. Their results were compared with the number of common subtopics shared by relevant document pairs as judged by editorial assessors. We utilized the correlation between judged similarity and measured similarity to evaluate the effectiveness of the selected similarity measures. Our intuition is that inter-document similarity is directly proportional to the number of common subtopics between document pairs.

1.3.2 Intent discovery

Our contribution in this area includes the discovery of various subtopics and query intents directly from text corpora. We uncovered diverse subtopics directly from text corpora using two methods. The first method mined diverse subtopics from the content of the top k relevant documents, while the second method utilized document anchors — anchor text and anchor links — to uncover diverse subtopics. Our anchor text approach for uncovering subtopics was implemented at the 2011 NTCIR INTENT subtopic mining task and the diversity task of the 2011 TREC Web Track.

Intent discovery from top k documents

Diversified query intents were uncovered from the top k documents. Using the *tf-idf* method, terms in the top k documents are selected, weighted, and ranked according to their importance in the document as well as the corpus. Terms having the highest scoring weights are selected and

clustered using the pointwise mutual information as their measure of association. Our work in this area (Akinyemi et al., 2010) has been published at the RIAO 2010 conference.

Intent discovery with document anchors

Anchor text and their corresponding out-links, i.e. URLs[4], are extracted from document collections. We extract the linkage information between an anchor text, its source document, and target document. We generated an anchor text linkage graph whose node is either an anchor text or a target document out-linked by an anchor text. The anchor text out-links to target documents form the edges of the graph. From a list of the terms in an anchor text, values representing a measure of association between the terms are obtained from the anchor text graph. By assuming that terms may be related if they appear in an anchor text that out-links a common page, even if the out-links originated from different pages, we may uncover related anchor text as well as related terms within the anchor text graph. By traversing the graph, closely related terms to given query terms are selected, weighted and ranked using their co-occurrence frequencies with query terms. Again, terms having the highest scoring weights are selected and clustered using pointwise mutual information as their measure of association.

Intent discovery from Chinese corpus

We implemented our anchor text subtopic mining technique on the 2011 NTCIR intent and subtopic mining task. The task organizers provided the SogouT corpus which consists of documents written in Chinese. From this corpus, we extracted anchor text and their source and target document links. The Chinese characters are encoded into their UTF-8 character encoding equivalent. We crudely segmented the UTF-8 representation of anchor text into their unigram and bigram tokens. Tuples of ⟨source document, anchor text, target document⟩ were considered as units of documents. A graph containing anchor text and their target documents was generated from the documents. The provided queries were also segmented into their unigram and bigram UTF-8 character encoding equivalents. Using a passage retrieval function (Clarke et al., 2006,

[4]uniform resource locator

2001) on the graph of anchor text and target documents, we retrieved anchor text and their outlink target documents using the given queries as input. For all retrieved target documents having additional anchor text edges, we further retrieve all the additional anchor text. Weights are assigned to individual anchor text and the weights are used to rank all the anchor text. The ranked anchor text set is clustered in order to eliminate duplicates and remove noisy anchor text from the anchor text set. Clusters of anchor text constitute diverse subtopics for the provided queries. Our work in this area has been published at the 2011 NTCIR conference on intent and subtopic mining task.

1.3.3 Diversity-aware retrieval models

We utilized terms in the subtopics mined from both the top k documents as well as the graph of anchor text and target documents as expansion terms used to expand the original query. Thereafter, we perform retrieval from the corpus based on the given queries. We consider the highest-ranked documents in the result as outputs of a diversity-aware ranking model. We increased diversity without compromising retrieval quality on our non-standard evaluation measure. Our contribution in this area includes showing that we can increase diversity without a significant drop in the quality of retrieval (retrieved documents).

Collection-based result diversification

We present a method that introduces diversity into document retrieval using clusters of top m terms obtained from top k retrieved documents. The query expansion terms are obtained using the method described in our intent discovery from top k documents method in Section 4.2. Terms from each cluster are used to automatically expand the original query. Our result indicates we can increase diversity without a significant compromise in the quality of retrieval. Our work in this area (Akinyemi et al., 2010) was published at the 2010 RIAO conference.

Diversification through anchor text links

We explore anchor text out-links as a method for diversifying search result. Our primary goal is the identification of aspects and interpretations of underspecified and ambiguous queries. To evaluate our approach, we obtain diverse query expansion terms from graphs of anchor text term co-occurrence frequencies created by our algorithm and apply them to a simple query expansion process. Using the TREC Web Track topics and the ClueWeb09[5] collection, we demonstrate that the expanded queries significantly improve novelty, as measured by standard measures. An implementation of the technique has been published at the diversity task of the 2011 TREC Web Track conference.

1.3.4 Diversity-aware evaluation models

Our contribution in this area include investigating alternative approaches for evaluating the performance of diversity-aware retrieval functions. As discussed earlier, since inter-document similarity correlates with subtopic judgment and subtopic categorization reflects diversity, we provide a model for evaluating the quality of diversity-aware ranking models. The evaluation makes use of measured inter-document similarity between pairs of relevant documents.

Do subtopic judgments reflect diversity?

Current measures of novelty and diversity in information retrieval evaluation require explicit subtopic judgments, adding complexity to the manual assessment process. In some sense, these subtopic judgments may be viewed as providing a crude indication of document similarity, since we might expect documents sharing common subtopics to be more similar on average than documents sharing no common subtopic, despite that the documents are relevant to the same topic. We test this hypothesis using documents and subtopic categorization done for relevant documents in the TREC 2009 Web Track. Result from our experiments demonstrate that document pairs having higher subtopic overlap also have higher inter-document similarity scores. This result provides additional validation for the use of subtopic judgment to measure novelty and

[5] `http://lemurproject.org/clueweb09/`

diversity, and point to new possibilities for measures of novelty and diversity. Our work in this area (Akinyemi and Clarke, 2011) has been published at 2011 ICTIR conference.

Diversity-aware evaluation based on inter-document similarity

We provide a model for evaluating diversity-aware ranking functions that does not rely on subtopic categorization of relevant documents. We have investigated the utility of inter-document similarity as a measure for evaluating the effectiveness of diversity-aware ranking functions. To the best of our knowledge, ours is the first model that incorporates inter-document similarity as an alternative for document subtopics information (or query intents).

1.3.5 Traceability

Our contribution in this area include the use of automatically generated soft links for discovering and maintaining traceable links. We provide an automatic, fast, robust and effective link discovery method for traceable links suitable for frequently edited documents such as source code and wiki applications. We provide various evaluation methods to test the effectiveness of our method.

Soft links: fast, effective and robust traceability links

Inspired by requirements traceability problems, we present a method for implementing fast and effective hypertext links to specific locations within documents. These soft links do not depend on tags, markups, or closed tool sets, yet they can generally survive extensive edits to a document collection, allowing the targets of these links to be located in real collections after years of frequent changes. We base our implementation of soft links on an existing passage retrieval algorithm, originally designed for question answering. The method treats the text surrounding the target of a soft link as a passage to be retrieved, creates a signature for that passage, and resolves the link by searching for the passage. The method is evaluated over a large collection of text and two large collections of source code, one written in C programming language and the other written in Java. Our findings and result have been published in the April 2012 edition of the *Software: Practice and Experience* journal.

1.4 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 provides some basic and background information about document similarity in general. Chapter 3 contains our work in the area of diversity, especially as related to inter-document similarity. The chapter also provides information about subtopic judgments carried out by editorial assessors in information retrieval evaluation frameworks. Details of our findings with respect to the correlation of subtopic judgment and measured inter-document similarity is also provided. The chapter concludes by describing our findings on the evaluation of inter-document similarity measures using subtopic judgment as the ground truth. We describe details of our approach towards evaluating diversity-aware ranking functions using inter-document similarity. Chapter 4 presents two subtopic mining and query intent discovery approaches devoid of user interaction data. These approaches are solely dependent on the document corpus. Chapter 5 presents our fast, effective and robust soft links approach for tracing frequently edited documents such as source code and Wikipedia. The chapter also contains how the soft link method is extensively evaluated. We conclude the thesis in Chapter 6 and present some future directions.

Chapter 2

Document Similarity

This chapter introduces document similarity. It begins with a discussion on the types and models of document similarity. This is followed by a general discussion on features used for computing document similarity and various approaches for representing these features. A general discussion on the definition of similarity follows models of document similarity. Next is a detailed discussion on measures of similarity suitable for measuring the similarity between document pairs.

2.1 Types of document similarity

There are different types of document similarity. Basically, the type of a document similarity implementation depends on the notion of similarity being modeled. Some document similarity types in the literature include: content-based similarity, similarity that is based on natural language processing (i.e., deeper linguistic features of documents.) Other types are feature-based similarity, directional similarity, perceived and judged similarity, absolute and relative similarity, and bounded and unbounded (i.e., normalized or non-normalized) similarity.

2.1.1 Content–based similarity

Content-based similarity uses hyperlinks, browsing patterns, document contents, and user interaction data, e.g. query and click logs to model similarity. According to Chen (1997), similarity between documents on the WWW can be inferred using their interrelationships which can be derived from their hypertext links, content similarity, browsing patterns or a combination of these methods. Hypertext linkage exploits link structures in web documents, such as incoming (*in-links*) and outgoing links (*out-links*). User interaction data such as query logs, click logs, download statistics and browsing history which constitute the browsing patterns of web users may be mined in order to discover features that are suitable for inferring document similarity.

2.1.2 Similarity based on Natural Language Processing (NLP)

Grefenstette (2009) categorized document similarity according to their syntactic, semantic and lexical properties. The syntactic property culminates from the underlying grammar and structure of texts in documents and how sentences are formed. NLP-based similarity utilizes one or a combination of syntactic, lexical or semantic properties of documents to quantify document similarity. The Semantic Web and WordNet thesaurus are examples of applications that incorporate NLP-based similarity of entities and terms. The semantic property comes from the meaning and interpretation of texts in documents, while the lexical property is a result of other deeper linguistic features such as word synonyms, antonyms, hypernyms and hyponyms. Obviously, the semantic feature is by far the most complex because it relates to deeper and deduced meaning of texts. For example, the phrase "*X has kicked the bucket*" means *an entity named X has died*. This type of interpretation can only be deduced by a person or a system that understands English language to the extent of knowing the meaning of the phrase. It is more difficult to automate this type of implied interpretation.

2.1.3 Feature–based similarity

Tversky (1977) decomposes object contents into features and postulates that the similarity between objects is proportional to their common features and inversely proportional to their distinct

features. Document features are predominantly the basis upon which document similarity is measured. Computing document similarity depends to a large extent upon how document features are represented. The vector space model (VSM) (Salton et al., 1975) relies on document features represented as term vectors in a multi-dimensional space. Feature representation in the latent semantic analysis (Deerwester et al., 1990; Landauer and Dumais, 1997) also follows the vector representation in the VSM.

2.1.4 Directional–based similarity

Tversky (1977) discussed symmetric and asymmetric similarity, i.e. directional similarity. In symmetric similarity, the measure of similarity between objects x and y denoted by $sim(x, y)$ is equal to the similarity between objects y and x, i.e., $sim(x, y) = sim(y, x)$. Symmetric similarity predominantly considers common features in objects. Distinct features contained by individual objects are largely ignored. The similarity between identical objects should always be symmetric. On the other hand, asymmetric similarity considers both the common and distinct features contained in each of the objects being compared. Therefore, $sim(x, y) = sim(y, x)$ iff $x = y$ and $sim(x, y) \neq sim(y, x)$ if $x \neq y$. According to Tversky, the similarity of Tel Aviv to New York differs from the similarity of New York to Tel Aviv, i.e. sim(*Tel Aviv, New York*) $\neq sim$(*New York, Tel Aviv*). In the former, i.e. sim(*Tel Aviv, New York*), if Tel Aviv is being compared to New York with respect to being a metropolitan city, it would suffice that New York may not be compared to Tel Aviv but cities having similar metropolitan features to New York or those having more population density than New York.

Tversky argues that the similarity between objects is directional; hence should generally be asymmetric. We can see that it is possible to evaluate similarity either as a product of common features in isolation or as a combined product of both common and different features of objects. Ashby and Perrin (1988) criticized this notion of asymmetric similarity by distinguishing between the "similarity of object x to object y" as being different from the "similarity between objects x and y".

2.1.5 Perceived and Judged similarity

Tversky (1977) discusses perceived and judged similarity. Perceived similarity is obtained through cognition that is learned from experiences. Judged similarity is either estimated or measured. Ashby and Perrin (1988) went further to distinguish between perceived similarity and judged similarity. Judged similarity is considered to be effective only when it corresponds to perceived similarity. On the other hand, if a similarity function estimates the similarity between two objects and the similarity judgment contradicts the perceived similarity, such function is deemed a sub-optimal similarity function. Santini and Jain (1999) expressed a transformation of perceived similarity into its judged similarity equivalent. If $s(x, y)$ represents the perceived similarity between features of objects x and y and $j(x, y)$ denotes the judged similarity, then

$$j(x, y) = g[s(x, y)], \tag{2.1}$$

where g is a monotonically non-decreasing function, s is based on perception, and j can be obtained experimentally.

2.1.6 Absolute and Relative similarity

Li et al. (2004) discussed absolute and relative similarity. Absolute similarity between two documents is the same irrespective of all the features contained in all other documents and every other phenomena that are external to the documents being measured. Similarity values between two documents is constant even if they are copied from one corpus to another. Other documents in the corpus do not affect the measure of association between the document pair. On the other hand, relative similarity provides similarity values that rely on other documents in the corpus. Similarity functions that rely on global corpus characteristics such as the inverse document frequency (*idf*) of terms compute relative similarity scores. If two documents are copied from one corpus to a different corpus, their measure of association will change between the two corpora.

2.1.7 Bounded and Unbounded similarity

Bounded and unbounded similarity is discussed by (Chen et al., 2007). Bounded similarity is constrained within a minimum and maximum values, usually between 0 and 1 where a similarity

value of 0 indicates document pairs that have nothing in common and a similarity value of 1 indicates identical documents. Unbounded similarity is unconstrained within a minimum or maximum values and the values can range between $-\infty$ and $+\infty$. Cosine similarity, frequency-based similarity (e.g., Dice coefficient and Jaccard index) are bounded, whereas the Kullback-Leibler (Kullback and Leibler, 1951) divergence measure of similarity in language models is unbounded.

2.2 Feature Representation

As previously stated, measures of document similarity is closely tied with the representation of document features. Next, we describe some common feature representation methods. These include the vector-based feature representation, word n-gram feature representation method, frequency-based representation, and network-based feature representation.

2.2.1 Vector space representation

The vector space representation model represents documents as vectors of terms and computes the similarity between document pairs by comparing their term vectors in a multi-dimensional plane. Term vectors could either be weighted or unweighted. The weighting usually combines both the term frequency (*tf*, for term importance in a document) and the inverse document frequency (idf, for collection-wide term importance) statistics. Documents that align more closely in the multi-dimensional plane are considered more similar. This alignment is calculated as the cosine of the angle between the term vectors.

2.2.2 Representation in language models

Language models have n-gram term representation, where $n \geq 1$. In the *unigram language model*, n=1, resulting in a bag-of-words model where the position of terms in the text is irrelevant. Apart from unigram term representation, higher values of n (i.e. $n > 1$), such as phrases

may also serve as units of term representation. Phrases or a combination of both unigram term and term phrases can be modeled as a unit of representation in language models.

2.2.3 Network-based representation

Network representations measure the distance between a pair of nodes on the network, where a node represents a document. Euclidean distance, Manhattan distance, or any other suitable distance calculation methods may be used to measure the distance between node pairs. Distance-based similarity measures are usually represented in networks (Rada et al., 1989) where nodes are objects whose similarity is being measured and edges are the distances between the objects.

2.2.4 Frequency-based representation

Another group of similarity measure employs the frequency-based representation. Normalized term frequencies are used to compute the similarity between documents. The intuition is to measure the similarity using the intersection of terms between document pairs, i.e., the common terms between them. Examples of frequency-based similarity measures include the Dice coefficient, Jaccard index, mutual information and pointwise mutual information (van Rijsbergen, 1979).

2.2.5 Representation in probabilistic models

In the probabilistic model, documents are represented as vectors of independent term features (i.e., term occurrence.) Sophisticated representations of the probabilistic model might assume term dependence. The ranking function utilizes the likelihood ratio of classifying a term as belonging to the set of non-relevant terms compared with the term being present in the set of relevant terms. For the binary independence model, values of term vectors are binary. The popular bm25 (Robertson et al., 2004) ranking function utilizes the feature representation in probabilistic models.

2.3 Definition of similarity

There is no *de facto* definition of similarity. However, the field of information theory has attempted to define similarity with respect to the commonality and differences between the objects in consideration. Variations of the commonality and differences between objects have been modeled severally to represent their similarity. Summarily, these methods either emphasize only the commonality, the differences, or a combination of the commonality and differences as descriptions and definitions of similarity. Mostly, similarity is expressed as the commonality between objects discounted by either their differences or a union of all features both objects contain.

In particular, Lin (1998) presented an information theoretic definition of similarity. The work attempts to provide a universal theoretical definition of similarity using axioms and assumptions that similarity functions should generally satisfy. It is obvious that the intuitions behind Lin's definition of similarity has been previously discussed by Tversky (Tversky, 1977). The intuitions are:

1. The similarity between objects x and y is related to their commonality. The more commonality they share, the more similar they are.

2. The similarity between objects x and y is related to the differences between them. The more differences they have, the less similar they are.

3. The maximum similarity between objects x and y is reached when x and y are identical.

Lin further formalized the intuitions using computable assumptions and axioms based on the three intuitions leading to the derivation of their Similarity Theorem which states that:

> The similarity between objects x and y is measured by the ratio between the amount of information needed to state the commonality of x and y and the information needed to fully describe what x and y are, such that:

$$sim(x, y) = \frac{f(x, y)}{f(x) + f(y)} \tag{2.2}$$

where *f(x)* represents the information encoded in x, *f(y)* denotes the information encoded in y, the operator $+$ represent various methods for combining $f(x)$ and $f(y)$, and *f(x,y)* represents the common information shared by both x and y. Even though object features are not specifically mentioned in the definition, information contents of objects are usually encoded and represented as features. Lin's work was implemented by Aslam and Frost (2003) specifically for document similarity using TREC's data set.

They observed that the information-theoretic method consistently and significantly outperforms other methods in their study for computing document similarity, such as Dice's coefficient, unweighted cosine and weighted cosine coefficients. Details of their evaluation method is provided in Section 3.3. We consider Lin's definition of similarity in Equation 2.2 as a generalized form of expressing the similarity between objects.

2.4 Models of Document Similarity

Models of document similarity depend on the representation of important features used for estimating the similarity between documents. The vector space model and language models of retrieval both utilize variants of vector space representation method. In this section we present models of similarity measures that utilize various representations of document features previously discussed. The similarity measures include vector space model, language models of retrieval, frequency-based similarity, similarity based on mutual information, and data compression ratio.

2.4.1 Vector Space Model

Salton et al. (1975) proposed the Vector Space Model (VSM) for document representation. Documents are represented as term vectors in a multidimensional plane. Weighted term frequencies are utilized to describe term importance in documents. A popular weighting approach is the *tf-idf* weighting technique where *tf* (term frequency) indicates term importance in a document and *idf* (inverse document frequency) quantifies term importance in the corpus. The similarity

between two documents is computed as the cosine of the angle between the term vectors of the two documents.

The cosine similarity between two documents represented as vectors d_x and d_y is given by:

$$Cosine(d_x, d_y) = \frac{\sum_{i=1}^{t} d_{xi} \cdot d_{yi}}{\sqrt{\sum_{i=1}^{t}(d_{xi})^2 \cdot \sum_{i=1}^{t}(d_{yi})^2}} \tag{2.3}$$

$$tf_{xi} = \frac{f_{xi}}{\sum_{k=1}^{n} f_{xk}} \tag{2.4}$$

$$idf_t = log\left(\frac{N}{N_t}\right) \tag{2.5}$$

where d_{xi} and d_{yi} represent weighted frequency (*tf-idf*) of the i^{th} term in d_x and d_y respectively. tf_{xi} represents the weighted term frequency component (for term i in document x) in the *tf-idf* weighting. f_{xi} is the frequency of term i in document x, n is the total number of terms in the document d_x. idf_t is the inverse document frequency portion of a term t in the *tf-idf* weighting. N represents the total number of terms in the corpus, and N_t is the total number of documents containing the term t in the whole collection. The cosine similarity is a normalized similarity metric and it is easy to compute. It is suitable for adhoc retrieval purposes even though other better performing ranking functions are now commonly used.

2.4.2 Language models of retrieval

The basic idea behind the language modeling approach is to regenerate a given reference document (D) from the language model (LM) of other documents in the corpus. Documents with language models more similar to that of D are considered more similar or relevant to D. The language model is used as a measure of similarity between document pairs. More sophisticated and smoothed language modeling approaches, such as the Kullback-Leibler divergence method, have been experimentally proven to outperform ranking functions based on the vector space model as a retrieval function. A major drawback of language models for ranking is that it does not directly

incorporate relevance in the model, but generates only the language model of the reference document D from the language model of other documents in the corpus (Lavrenko and Croft, 2001; Jones et al., 2003; Kurland, 2006). Lavrenko and Croft (2001) addressed this problem of indirect relevance modeling in language models by establishing a relation between the language models and probabilistic models.

Language models of retrieval have been investigated substantially and found to be relatively effective. As an added benefit, it has a sound statistical and probabilistic foundation. Given two documents D and d where D is the reference document such that $D = \langle t_1^D, t_2^D, ..., t_m^D \rangle$. The terms in D are represented as $\langle t_1^D, ..., t_m^D \rangle$ and terms in d are represented as $\langle t_1^d, t_2^d, ..., t_n^d \rangle$. The similarity between D and d is taken as an estimation of generating the document D from the language model (Ponte and Croft, 1998) of d, i.e. $p(D|M_d)$, where M_d is the language model of d and p is a probability function. A very simple probability estimate common in language modeling is the maximum likelihood probability estimate. The maximum likelihood model $M_d^{ml}(t)$ for term $t \in d$ is computed as:

$$M_d^{ml}(t) = \frac{f_{t,d}}{|d|}, \tag{2.6}$$

where $f_{t,d}$ is the frequency of term t in document d and $|d|$ is the total number of terms in d. Term independence is assumed and the similarity function $sim(D|d)$ between the documents d and D is given by:

$$sim(D|d) = \prod_{i=1}^{m} p(t_i^D|d). \tag{2.7}$$

This similarity, i.e., $sim(D|d)$ is equivalent to $p(D|d)$ which may be used as a ranking function.

$$p(D|d) = \prod_{t \in D} M_d(t). \tag{2.8}$$

Collection–wide information (*cwi*) may be used for *smoothing* terms having zero frequencies in the query in order to avoid the zero probability or data sparsity problem such that the *cwi* for term t, *cwi(t)* is given by:

$$cwi(t) = \frac{f_{t,C}}{|C|}, \tag{2.9}$$

where $f_{t,C}$ is the collection-wide frequency of term t and $|C|$ is the total number of terms in the corpus. Next, we describe the Kullback-Leibler divergence which is a divergence method – based on the language model – used for computing the similarity between documents.

Kullback-Leibler Divergence (KLD):

A very common approach is to estimate the divergence or the relative entropy between the probability distributions of terms in the reference document D and terms in other documents. Documents are subsequently ranked based on the divergence of their terms distribution from the terms distribution of D. This method is known as the Kullback-Leibler divergence (KLD) (Kullback and Leibler, 1951; Büttcher et al., 2010; Croft et al., 2009) and it is given by:

$$kld(D, d) = \sum_{t \in D} M_D(t) \cdot log \frac{M_D(t)}{M_d(t)}, \tag{2.10}$$

where $M_D(t)$ and $M_d(t)$ are distributions of term t in the reference document D and another document d respectively.

Kurland (2006), Kurland and Lee (2004), and Meister et al. (2010) explored the application of language models for estimating inter-document similarity. Their work combines document-based language models with additional inter-document similarity information. They obtained their inter-document similarity information from overlapping clusters of documents in the corpus. They represented the corpus both as a set of documents as well as a set of clusters of similar documents. They combine the language models of both documents and clusters such that if a document does not contain the terms in their reference document (which is a query), but belong to the same cluster of a relevant document, the document can still be considered relevant to the query.

The inter-document similarity scores obtained from the clusters using KLD are plugged as an additional parameter into their ranking function. Relevant documents that appear less relevant

to the reference document might still be ranked high because they belong to the same high-ranking cluster which contains other relevant documents. Consequently, clusters become useful for selecting relevant documents such that documents that belong to relevant clusters are ranked higher. The clusters are query independent and for efficiency, they are created offline. They reported a more effective ranking function producing higher recall and precision at higher ranks than the basic language modeling approach.

J-Divergence (Symmetric KLD):

The J-divergence is a symmetric version of the KL-divergence. The J-divergence J between documents d_i and d_j is the average of $KLD(d_i, d_j)$ and $KLD(d_j, d_i)$. Equation 2.11 shows the formula for computing the J-divergence between two documents.

$$J = Average(KLD(d_i, d_j), KLD(d_j, d_i)) \tag{2.11}$$

Apart from the symmetric property of J-divergence, it has not been investigated enough to arrive at a definite conclusion with respect to its effectiveness when compared to KLD.

2.4.3 Frequency-based similarity

Frequency-based similarity models utilize the frequency of document terms for estimating the similarity between two documents. Dice coefficient (Dice, 1945) and Jaccard index (Real and Vargas, 1996) are examples of frequency-based similarity models. The Dice coefficient is computed with Equation 2.12 and the Jaccard index is computed with Equation 2.13.

$$Dice(d_i, d_j) = \frac{2|d_i \cap d_j|}{|d_i| + |d_j|} \tag{2.12}$$

$$Jaccard(d_i, d_j) = \frac{|d_i \cap d_j|}{|d_i \cup d_j|} \tag{2.13}$$

where d_i and d_j represent documents i and j, $|d_i \cap d_j|$ represents the frequency of common terms contained in both d_i and d_j, and $|d_i \cup d_j|$ represents the number of the terms contained in the union of d_i and d_j.

2.4.4 Mutual information–based similarity

The similarity model based on mutual information such as pointwise mutual information (pmi) utilizes the co-occurrence information of document features to estimate document similarity.

$$pmi(d_i, d_j) = log \frac{|d_i \cap d_j|}{|d_i| \cdot |d_j|} \tag{2.14}$$

Equation 2.14 shows the formula for computing the pmi between two documents. The frequency of common terms in the two documents is normalized with the product of their individual term frequencies. The resultant normalized ratio is transformed with the logarithm function.

2.4.5 Data compression ratio–based similarity

The data compression (Cilibrasi and Vitányi, 2005) ratio of an information is based on the transformation of the bit rate encoding of the data in a document to that of another document. The bit rate required to transform a document into another document is utilized as a similarity measure. For a pair of documents d_i and d_j, we may obtain the data compression rate of encoding each document to the terms the documents have in common, i.e. $(d_i \cap d_j)$. The data compression ratio between d_i and $(d_i \cap d_j)$ is represented as r_i. Likewise, r_j represents the data compression ratio between d_j and $(d_i \cap d_j)$. Equations 2.15 and 2.16 show the computation formulas for both r_i and r_j respectively.

$$r_i = \frac{|d_i \cap d_j|}{|d_i|}, \tag{2.15}$$

$$r_j = \frac{|d_i \cap d_j|}{|d_j|}. \tag{2.16}$$

The data compression ratio-based similarity measure combines r_i and r_j. The similarity score between documents d_i and d_j should be *zero* when they have nothing in common and *one* when they are identical documents. Various combinations of the data compression ratios are shown in Equations 2.17 to 2.20.

$$CRAvg(d_i, d_j) = \frac{r_i + r_j}{2} \tag{2.17}$$

$$CRProd(d_i, d_j) = r_i \cdot r_j \tag{2.18}$$

$$CRMin(d_i, d_j) = r_i \cdot r_j \cdot Min(r_i, r_j) \tag{2.19}$$

$$CRMax(d_i, d_j) = r_i \cdot r_j \cdot Max(r_i, r_j) \tag{2.20}$$

Equation 2.17 shows the average of the two ratios, Equation 2.18 utilizes the product of the two ratios, Equation 2.19 combines the product of the data compression ratios with their minimum ratio value, while Equation 2.20 combines the product of the ratios with their maximum data compression ratio value.

2.5 Similarity Measures

In this section, we describe three applications of document similarity measures in information retrieval. First, we discuss the application of similarity measures for quantifying the similarity between a query and documents. The resulting query–to–document similarity measure constitutes the foundation of adhoc retrieval and ranking models when a user's information need is represented as queries and the output of the retrieval task is a ranked list of relevant documents. Next, we describe the application of similarity measures for quantifying the similarity between document pairs. The resulting document–to–document similarity measure addresses the information need when a direct measure of association between two given documents is required.

We conclude the section by describing the query–sensitive similarity. Query–sensitive similarity measures apply to situations when both query–to–document and document–to–document similarity measures are combined for improving the effectiveness of either the retrieval task (in query–to–document similarity) or inter-document similarity in document–to–document similarity.

2.5.1 Query–to–Document similarity

Query–to–document (Q–D) similarity measures are also called retrieval functions. Examples include the probabilistic binary independence model of retrieval of which the popular *bm25* (Robertson et al., 2004) ranking algorithm is built upon, KL-divergence that is built upon the language models of retrieval, and learning to rank document ranking algorithms that are based on machine learning techniques. All these similarity (or ranking) methods take as input a user information need in form of a query and produce a list of relevant documents from the corpus ranked according to the probability ranking principle (van Rijsbergen, 1979) which states that:

> if an information retrieval system's response to each query is a ranking of the documents in the collection in order of decreasing probability of relevance, the overall effectiveness of the system to its user will be maximized.

2.5.2 Document–to–Document similarity

Document–to–document (D–D) similarity computes a measure of association between two documents. The similarity function takes as input two documents and provides a score representing the similarity between the document pair. Examples include the cosine coefficient, Dice coefficient, Jaccard index, data compression ratio of documents, and others.

Query–to–Document vs. Document–to–Document similarity

There are fundamental differences between Q–D similarity measures and D–D similarity measures. First, they have different input parameters. Q–D input parameters consist of a query and a corpus that contains documents, whereas D–D input parameters are two documents. They also have different outputs. Documents in the corpus are compared against the provided query in order to ascertain a relevance (or similarity) score for the document with respect to the query. Relevance scores of documents are sorted in descending order and used to rank the documents. A *list* of the topmost documents having the highest scores are returned to the user. However, the output of a D–D similarity is a *value* of the similarity score representing a measure of closeness between the two documents in consideration.

Actual measure of similarity is very important in D–D similarity; hence an actual measure of association between document pairs is directly incorporated into inter-document similarity models such that an inter-document similarity score represents an actual measure of similarity between document pairs. Inter-document similarity scores between retrieved documents in a Q–D retrieval task is also important, but it is basically used as a means to rank the list of relevant documents in the result. A notable study carried out by Diaz (2007) extends the application of the cluster hypothesis to include the notion that closely related documents should also have similar similarity scores. Diaz provided an optimization technique that regularizes retrieval scores of documents that are topically relevant to a query or that have high inter-document similarity scores. He reported an improvement on retrieval effectiveness — using the MAP effectiveness measure — when compared with baselines of some standard retrieval models such as the Okapi vector space model and language models of retrieval.

In this work, we focus on D–D similarity. We explore the foundational models retrieval functions are based upon. We disregard assumptions that are peculiar to Q–D similarity as we explore other assumptions deemed more appropriate for D–D similarity. Throughout this thesis, we make clear when we discuss Q–D similarity by specifically mentioning the presence of a query, whereas for D–D similarity discussions, we only mention documents and no query.

2.5.3 Query–Sensitive similarity

Query–sensitive similarity cuts across both the Q–D and D–D similarity measures. The goal here is to compute measures of similarity between document pairs while the similarity measure is biased according to a given query. It is also possible to utilize inter-document similarity as a method to improve the effectiveness of Q–D ranking functions (Diaz, 2007; Kurland, 2006).

2.6 Similarity metrics

Some similarity measures qualify as similarity metrics while some don't. There is a distinction between similarity metrics and similarity measures. If *d(x,y)* stands for the distance between objects x and y when x is the reference point and *d(y,x)* denotes the distance between objects y

and x when y is the reference point. A metric has been described in the literature (Chen et al., 2007; Tversky, 1977; Ashby and Perrin, 1988; Santini and Jain, 1999; Rada et al., 1989) as a measure that satisfies the following additional properties:

- Self-similarity property: $d(x, x) = d(y, y)$,
- Zero property: $d(x, x) = 0$,
- Positive property: $d(x, y) \geq 0$,
- Symmetry property: $d(x, y) = d(y, x)$ vs. asymmetry $d(x, y) \neq d(y, x)$,
- Triangle inequality property: $d(x, y) + d(y, z) \geq d(x, z)$, unfortunately, the triangle inequality property does not always hold.
- Minimality axiom property: $d(x, y) \geq d(x, x)$

Transforming a bounded distance function having lower bound 0 (for identical documents) and upper bound 1 (for documents having nothing in common) into a similarity function $sim(x, y)$ is easily accomplished with the expression in Equation 2.21.

$$sim(x, y) = 1 - d(x, y). \tag{2.21}$$

Such similarity function also has the [0,1] bound in which case if $sim(x, y)$ is 0, then the documents have nothing in common; and if $sim(x, y)$ is 1, then the documents are identical. Some similarity measures, such as the KLD, do not qualify as similarity metrics because they violate some of the metrics axioms. For example, the KLD violates both the *zero* and *symmetry* properties of a metric.

KLD scores are usually negative numbers that are less than zero. Consider the KLD scores k_a and k_b for two pairs of document pairs (d_a^i, d_a^j) and (d_b^i, d_b^j) respectively. If $k_a > k_b$, then a similarity metric should imply that d_a^i is more similar to d_a^j than d_b^i is to d_b^j. However, KLD does not provide this guarantee. The cosine coefficient, frequency-based similarity methods, and the data compression ratio similarity measures mostly satisfy the metrics properties.

Chapter 3

Similarity and Diversity

This chapter is sub-divided into three parts. First, we introduce the concept of diversity and novelty in information retrieval. As a result of the predominance of short queries used for ad-hoc search, diverse user information needs are described with ambiguous and under-specified queries. Result diversification has been the approach used to address this problem. Apart from query ambiguity, information retrieval systems also strive to satisfy user information needs more effectively by providing new and fresh information at lower ranks. They do this while reducing or completely eliminating information redundancy, i.e., a repetition of information that has been seen at higher ranks, at lower ranks. Information retrieval tasks that cater for this category of retrieval and ranking problems are discussed in this chapter. We explore retrieval and ranking models that support such novelty and diversity–aware retrieval and ranking. We also discuss models that are used to evaluate the effectiveness of diversity–aware retrieval and ranking functions.

In the second part, we investigate the effectiveness of human involvement in the process of evaluating diversity–aware retrieval and ranking functions using inter-document similarity. As a next step and in our quest to explore how to evaluate diversity–aware retrieval and ranking functions devoid of explicit human involvement, we consider the utility of inter-document similarity. This prompted our investigation to evaluate the effectiveness of inter-document similarity measures in order to determine which one to utilize in our experiments. Our finding in evaluating the performance of inter-document similarity measures is also presented in this chapter. In the third

part of the chapter, we conclude by providing an evaluation model that measures the effectiveness of diversity–aware retrieval and ranking functions based on inter-document similarity.

3.1 Diversity

Search users often provide ambiguous and underspecified queries. For the query *windows*, some users might be satisfied with information about glass windows, the Microsoft Windows operating systems, others might be interested in only replacement windows, X Windows, Windows musical group, or the Windows movie. How can the search engine satisfy these diverse user intentions that are represented with the ambiguous query *windows*?

Predicting the search intention (user intent) of search users when a query is ambiguous is a difficult problem. Since the same search engine serves several users having varied intentions, the problem of deciding the correct intention of a user's ambiguous query is challenging. Different users have different interpretations of the same query. Therefore, a solution that has been actively explored in the literature is result diversification (Carbonell and Goldstein, 1998; Akinyemi et al., 2010; Clarke et al., 2008; Santos et al., 2010b; Radlinski et al., 2010; Carterette and Chandar, 2009). When a search result is diversified, the result reflects a variety of possible intentions for the query. In our example, the search engine could retrieve documents that satisfy each of the known possible intents and hope most users will be satisfied having their information need within the top-k of the ranked result.

Searching involves an active interaction between two major entities: A *user* presumably having an information need that is represented as a query and a text collection providing the user information need in form of the search result. In order to detect and derive diversified search result, Web search engines tune their retrieval functions to cater for result diversification. They make use of user interaction data such as query log, click log which may also contain abandoned result (i.e. result that are not acknowledged with a click by end users) to simplify the process of result diversification. By considering previous user behavior, subsequent results may be skewed to match the derived intentions of most search users. This approach is particularly suitable for Web search engines because of their size and the diversity of their users. However, the same approach may be unsuitable for other search categories, such as enterprise and vertical search in

which case the number of search users is smaller and there is limited inherent diversity in the documents.

Another direction taken in prior work is to use organized data sources and general world knowledge such as the disambiguation annotations in Wikipedia[1] as well as the taxonomy of entities or thesaurus in WordNet[2] to identify various interpretations of a query (Hu et al., 2009; Rafiei et al., 2010; Clough et al., 2009; Santamaria et al., 2010). Wikipedia disambiguation pages provide pointers to diverse interpretations of queries and terms. Despite this, there is no guarantee that search users prefer the generalized query interpretations that utilize world-knowledge as provided by these systems. There is also no guarantee that the interpretations provided describe statistical distributions of topics in the collection. In another example, the query "*obama*" on the Wikipedia disambiguation page has more than ten interpretations, with only one referring to the U.S. President Barack Obama and his family. Documents returned for the same query on a popular commercial search engine are predominantly skewed towards the president. As an additional constraint, WordNet does not provide support for disambiguating proper nouns.

It has been suggested that the clarity of ambiguous queries can be improved by adding more terms (Cronen-Townsend and Croft, 2002) to such queries. The goal of diversity in information retrieval is to accommodate the variety of user needs underlying a query, since different users may use the same terms to mean different things. This problem is attracting growing attention from the research community, and there exists a number of re-ranking approaches that have been proposed to introduce diversity into search result. Next, we present some related work in the area of diversity–aware retrieval and ranking models.

3.1.1 Diversity–aware retrieval

Query reformulation and expansion have been studied and used for result diversification (Santos et al., 2010a; Dang and Croft, 2010; Akinyemi et al., 2010). Akinyemi et al. (2010) provided an explicit result diversification that relies on query expansion with the expansion terms derived from subtopics covered by a given query. Terms in an expanded query are used for document

[1] `www.wikipedia.org`

[2] `wordnet.princeton.edu`

retrieval and the result is combined in a round-robin manner. Santos et al. (2010a) utilized sub-queries covering aspects of an original query as a means to re-rank retrieval result in order to infuse diversity into the initial ranked result. They made use of a probabilistic approach that caters for document relevance and subtopic coverage such that both properties are maximized.

Carbonell and Goldstein (1998) proposed a re-ranking method based on a *Maximal Marginal Relevance* (MMR) criteria. Their method aims to strike a balance between the relevance of a document with respect to a query, and the similarity between a candidate document (being considered for inclusion into the ranked result) and previously selected documents at higher ranks. Their algorithm proceeds by systematically and simultaneously selecting documents that are relevant to a given query, but also differ from previously selected documents at higher ranks.

Zhai et al. (2003) built language models to estimate two similarity metrics related to the MMR criteria, i.e., inter-document similarity and retrieval function. They proposed methods to diversify results so as to cover different *subtopics* for a given query. Chen and Karger (2006) explored the problem of query ambiguity in Web search using a negative feedback approach. By treating higher ranked documents as *non relevant* documents, they attempt to diversify result by selecting documents that are not similar to higher-ranked documents.

Carterette and Chandar (2009) provided faceted-based topic retrieval ranking models for diversifying retrieved top k documents. One of their models prunes top k documents by calculating the similarity between the documents and the given query and removing subsequent documents having a similarity score beyond a certain threshold. Their other model is based on a probabilistic model that attempts to retrieve a small set of documents having a high probability of covering diverse facets most relevant to the query. He et al. (2011) explored diversity–aware retrieval and ranking based on clusters of relevant documents. They proposed that relevant documents may be clustered. After which the clusters of relevant documents may be ranked. The top k documents in high-scoring clusters may then be selected in a round-robin manner.

On the Web, Radlinski et al. (2010) have utilized query and user interaction logs to provide a diversity–aware ranking model that rewards diversity and punishes redundancy. Query logs are the main source for uncovering query intents (Ashkan et al., 2009; Baeza-Yates et al., 2006; Jansen et al., 2007; Rafiei et al., 2010; Jansen et al., 2008; Song et al., 2009; Herrera et al., 2010; Clough et al., 2009). An implicit assumption in using query and click logs for uncovering intents

of queries is that users will only (or mostly) search using previously used query terms. But, it is not clear whether this assumption always hold. Again, such logs are not widely available outside commercial search engine companies. Some learning based methods have also been proposed to diversify results on the Web (Radlinski et al., 2008; Yue and Joachims, 2008).

Zhai and Lafferty (2006) proposed the risk minimization and portfolio theory for modeling retrieval functions. Given a query and reformulated versions of the query which altogether produced various ranked list of relevant documents. They incorporated a probabilistic approach that models a user's choice of selecting a list of relevant documents as a decision making task. The model contains a loss function that estimates a risk value for each of the possible strategies of document list selection. The goal is to minimize the risk. They provided a retrieval function suitable for subtopic retrieval in which case document relevance is assumed to be dependent on other relevant documents at higher ranks. Wang and Zhu (2009) as well as Rafiei et al. (2010) also provided portfolio-based retrieval models in which case document relevance is not in isolation but computed as being dependent on other relevant documents at higher ranks. The utility of the whole ranked list is maximized in order to cover more aspects of the query.

3.1.2 Diversity–aware evaluation models

On diversity-aware evaluation models, Clarke et al. (2008) described α-nDCG — a nugget-based evaluation measure — based on normalized discounted cumulative gain (nDCG) (Järvelin and Kekäläinen, 2002). Like nDCG, α-nDCG uses a graded relevance-based evaluation framework. It rewards the existence of diversity and novelty in top-k retrieved documents by penalizing redundant information. Agrawal et al. (2009) based their approach on the existence of intent based taxonomy that determines the probability of a user being interested in a given intent. They proposed methods to maximize the intents covered by the retrieved results. Their work is incorporated into the intent aware (IA) evaluation models for diversity.

Chapelle et al. (2009) proposed the expected reciprocal rank (ERR) evaluation model for graded relevance. Apart from the positions of ranked documents, the relevance of documents appearing at higher ranks are also factored into the ERR model. ERR attempts to model the satisfaction (*gain*) a user enjoys from the document at each rank with respect to the satisfaction from previously seen documents at higher ranks. The intent aware version of ERR (i.e., ERR-IA)

is in use for evaluating diversity–aware ranking models at the 2010 and 2011 diversity task of the TREC Web track. Sakai and Song (2011) proposed the D and $D\sharp$-measures which reward relevant documents containing more popular intents and penalize those having less relevance and marginal intents. D-measures are similar to the family of Agrawal's intent aware measures but with a different normalization. $D\sharp$ is a D-measure that is biased to reward ranked lists having high intent recall at a particular rank.

In the next section, we investigate the quality of subtopic judgment categorization carried out by human editorial assessors for diversity–aware retrieval tasks in the TREC evaluation framework. We present results of our findings investigating whether subtopic judgments actually reflect diversity as measured by inter-document similarity.

3.2 Do subtopic judgments reflect diversity?

Current measures of novelty and diversity in information retrieval evaluation require explicit subtopic judgments, adding complexity to the manual assessment process. In some sense, these subtopic judgments may be viewed as providing a crude indication of document similarity, since we might expect documents relevant to common subtopics to be more similar on average than documents sharing no common subtopic, even when these documents are relevant to the same overall topic. In this section, we test this hypothesis using documents and judgments drawn from the TREC 2009 Web Track. Result from our experiments demonstrate that higher subtopic overlap among relevant documents correlates with higher inter-document similarity. Our result provides a validation for the use of subtopic judgments and pointing to new possibilities for measuring novelty and diversity.

3.2.1 Introduction

Several ongoing information retrieval evaluation efforts, including the TREC Web Track[3] and the NTCIR Intent Task[4] focus on the evaluation of novelty and diversity–aware ranking models.

[3] `plg.uwaterloo.ca/~trecweb`
[4] `www.thuir.org/intent/ntcir9`

For both the TREC Web and NTCIR Intent tracks, each evaluation topic is structured around a typical Web query. A number of subtopics are defined for the query in each of the topics, with each subtopic reflecting a distinct aspect or interpretation of that query. For example, subtopics associated with the query "tornadoes" (topic 75 in the TREC Web Track) address their causes, occurrences, forecasting, and fatalities, as well as requesting images and videos. Prior to submitting their experimental runs, participants are given a collection of Web documents and a set of queries, but not the subtopics associated with the queries.

For each query in the Web track, participants attempt to infer the diversity underlying the query and return a ranked list of documents that balances novelty against relevance (Clarke et al., 2011). After submission, assessors judge each document independently with respect to each subtopic. Results are reported using measures designed to evaluate novelty and diversity, such as α-nDCG (Clarke et al., 2008), ERR-IA (Chapelle et al., 2009), and "intent aware" versions of traditional measures (Agrawal et al., 2009), all of which depend upon the availability of subtopic judgments.

The NTCIR Intent task require participants to submit a list of subtopics they could uncover for each given query. In a similar manner to the Web track, assessors also judge the result submitted by the task participants with respect to the quantity and quality of the subtopics submitted for each query. The organizers reported performance results using the $D\sharp$ measure (Song et al., 2011). A significant difference between the TREC Web track and NTCIR Intent and subtopic mining tasks is that the TREC Web track requires document retrieval whereas the subtopic mining task requires subtopics only, i.e., document retrieval is not included.

We investigate the relationship between measured inter-document similarity and the subtopic judgments rendered by the assessors for the Web track. If these judgments genuinely reflect diversity, the average similarity between documents relevant to same subtopic should be higher than the average similarity between documents that are relevant to different subtopics. By testing this hypothesis, we seek to provide validation for the use of subtopics as a measure for novelty and diversity in information retrieval evaluation. In addition, we hope to lay the groundwork to augment or replace explicit subtopic judgments with measured inter-document similarity values. Our result provides a basis for using measured document similarity as an alternative to subtopic-by-subtopic judgments of relevant documents.

We see an obvious connection between our hypothesis and the venerable cluster hypothesis (Hearst and Pedersen, 1996; Voorhees, 1985; Smucker and Allan, 2009; van Rijsbergen, 1979), which states that "closely associated documents tend to be relevant to the same requests" (van Rijsbergen, 1979). We extend the idea to the subtopic level, but with a focus on the relationship between the documents that are relevant to a given query, resulting in our diversity-oriented cluster hypothesis discussed in Section 1.2.2. We expect that documents that are relevant to the same subtopic will tend to be more closely associated than documents that are relevant to different subtopics.

In the next two sub-sections, we experimentally investigate our hypothesis. For each pair of documents that are relevant to a given query, we consider the degree of *subtopic overlap* between them, i.e. the number of subtopics for which both documents are relevant. We then compare this overlap against the traditional cosine inter-document similarity function. In effect we treat subtopic overlap between relevant documents as a crude similarity value. As the basis for our experiments, we use the topics and judgments from the TREC 2009 Web Track (Clarke et al., 2009).

3.2.2 Method

We start with the `qrels` file provided by TREC 2009 Web Track's diversity task, which encodes the judgments for the task. This file contains a list of tuples, each composed of four fields: document id, topic number, subtopic number, and relevance judgment. Each tuple indicates that the given document is either relevant or not relevant to the given subtopic of the given topic. While the `qrels` file includes both relevant and non-relevant judgments, we exclude documents that were not judged relevant to at least one subtopic, since we aim to compare similarity between pairs of relevant documents.

For each topic, we consider all pairs of documents relevant to at least one of the subtopics of that topic. For each pair, we compute two values: 1) standard cosine similarity, and 2) subtopic overlap, which indicates the number of relevant subtopics shared by the two documents. For the TREC 2009 Web Track documents, the subtopic overlap values range from 0 to 4, with most document pairs having subtopic overlap values of 0, 1 or 2. Next, we present the result from our experiments.

3.2.3 Results

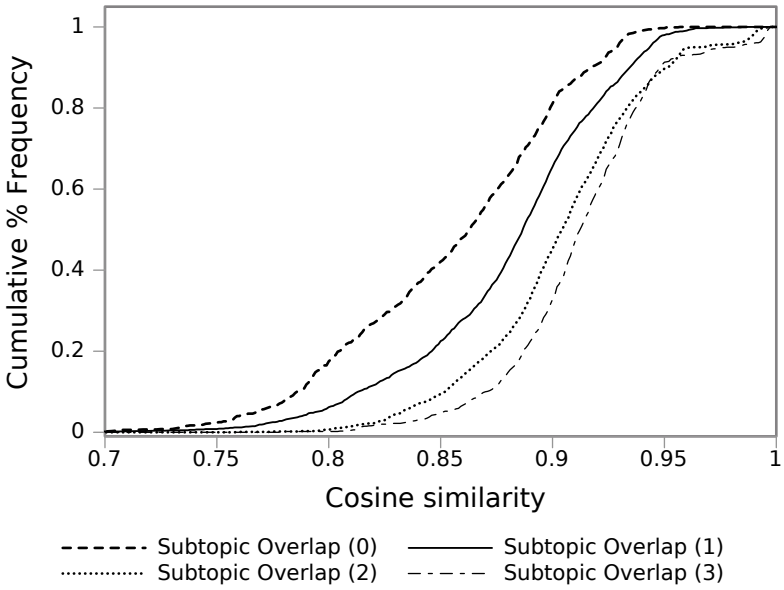
For our analysis, we focus on document pairs having subtopic overlap values of 0, 1 and 2, because only a very small number of document pairs have subtopic overlap values of 3 or 4. Our hypothesis suggests that larger cosine similarity values should correlate with larger subtopic overlap values. To compare these values, we compute the distribution of the cosine values for each topic with respect to the cumulative percentage frequency of their subtopic overlap values.

The plots in Figures 3.1 to 3.4 show the distribution of cosine similarity values for document pairs with different levels of subtopic overlap for eight example topics. Each curve provides a cumulative distribution for a given level of subtopic overlap, where a specific point on the curve indicates the percentage of pairs with cosine values less than or equal to that value. All eight examples support our hypothesis, with document pairs having higher subtopic overlap values consistently having higher cosine similarity values. For example, in Figure 3.4(a) more than 80% of pairs with overlap 0 have cosine similarity values falling below 0.95, while more than 65% of pairs with overlap 2 have values above 0.95.

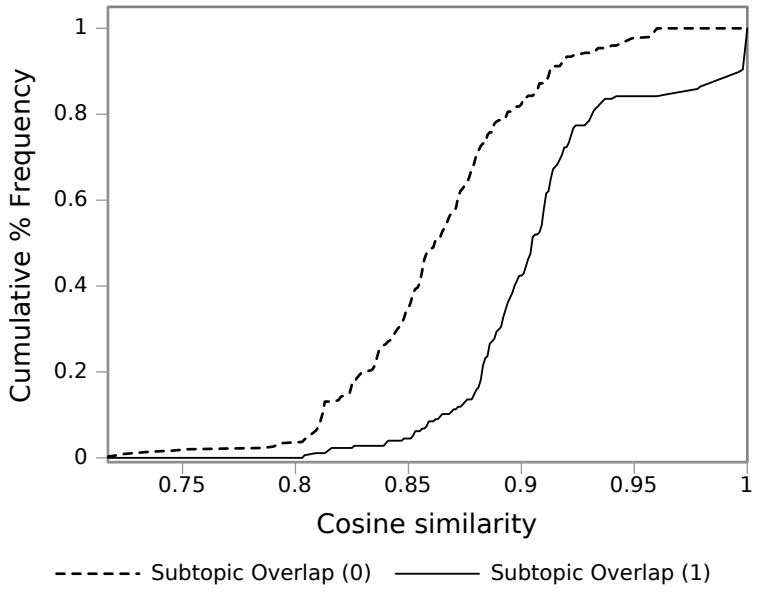
Plots for most other topics follow the same trend, supporting our hypothesis and providing validation for the use of subtopics as an indicator of novelty. For each TREC 2009 topic, we calculated the mean cosine similarity for document pairs with different overlap values. For 86% of TREC 2009 topics, documents pairs with overlap > 0 exhibited higher mean similarity than documents with overlap 0. For example, for topic 10, pairs with overlap 0 have a mean cosine similarity of 0.855, while pairs with overlap 1 have a mean cosine value of 0.880 and pairs with overlap 2 have a mean cosine value of 0.903. As a statistical test, we computed a paired t-test across the topics, comparing different levels of overlap. All p-values are $\ll 0.01$.

3.2.4 Summary

We have demonstrated that document pairs having overlapping subtopics also tend to have higher similarity values when measured by standard cosine inter-document similarity measure. This result provides validation and support for the use of subtopic judgments to measure novelty and diversity in information retrieval evaluation. In the future, we hope to extend our experiments to other similarity measures and test collections.

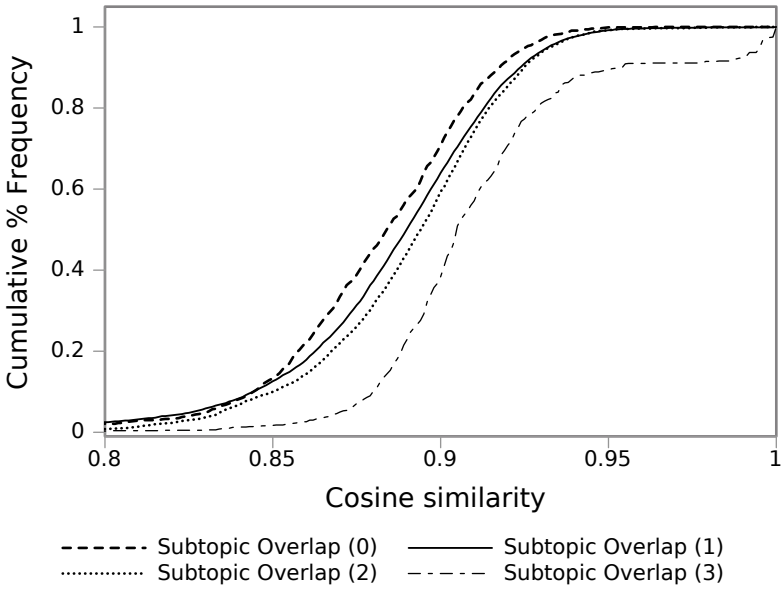


(a) Topic 10

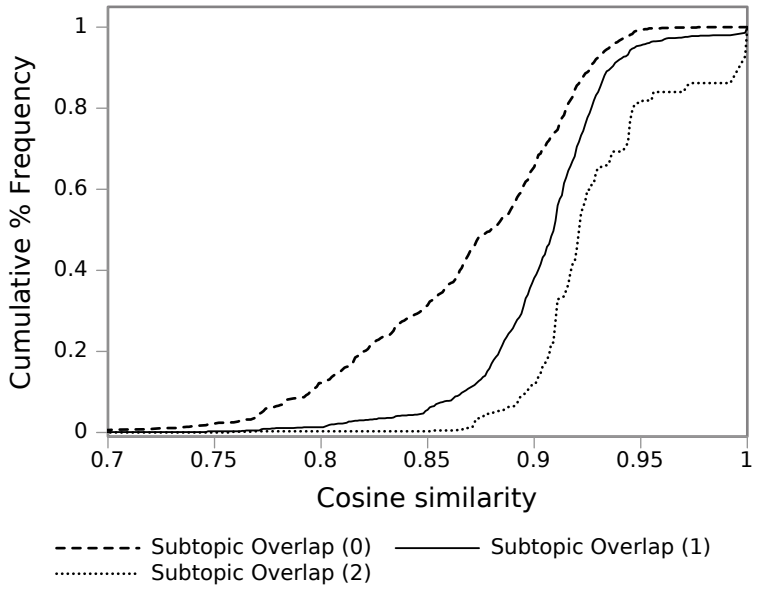


(b) Topic 16

Figure 3.1: Distribution of cosine similarity values for topics 10 and 16.

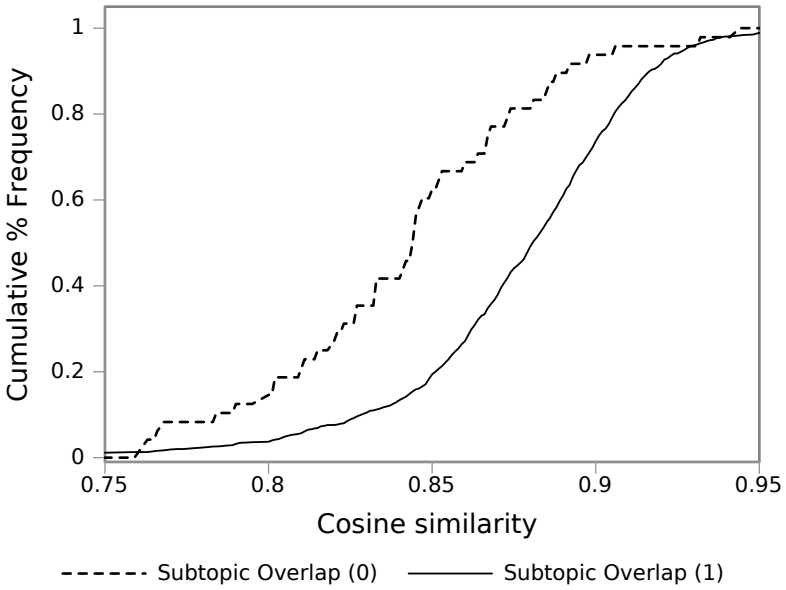


(a) Topic 26

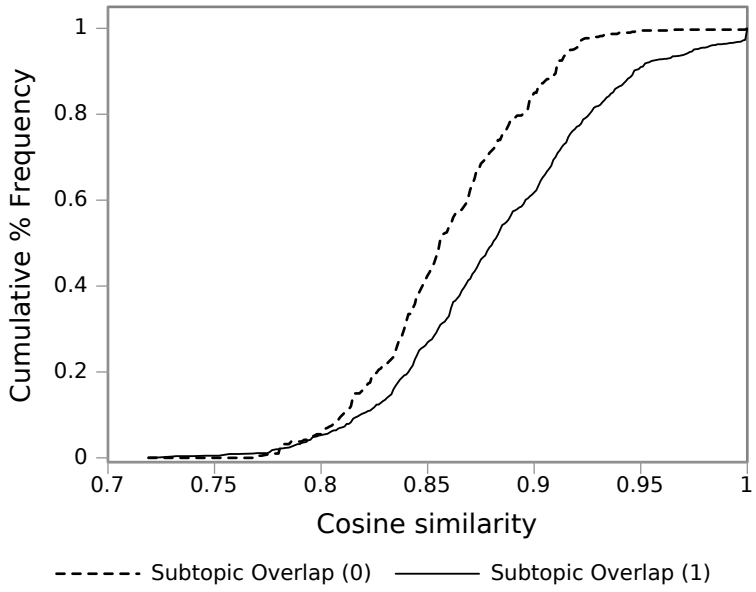


(b) Topic 31

Figure 3.2: Distribution of cosine similarity values for topics 26 and 31.

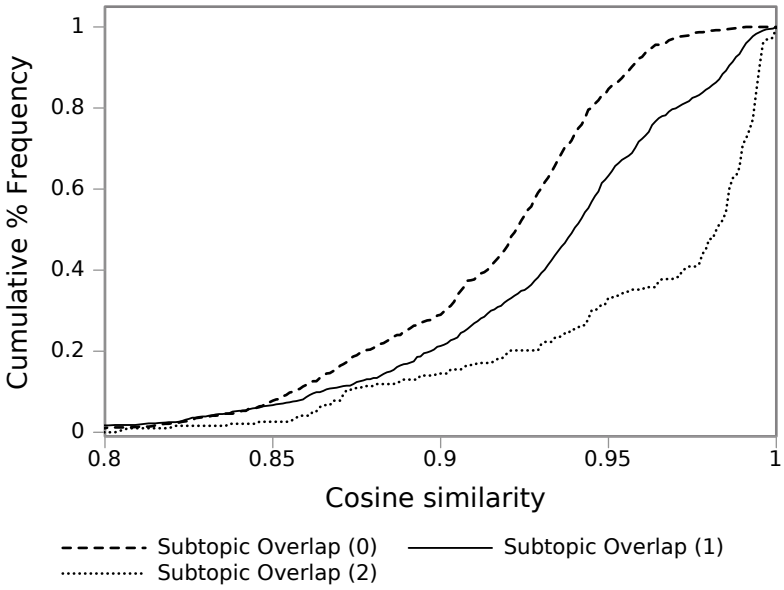


(a) Topic 36

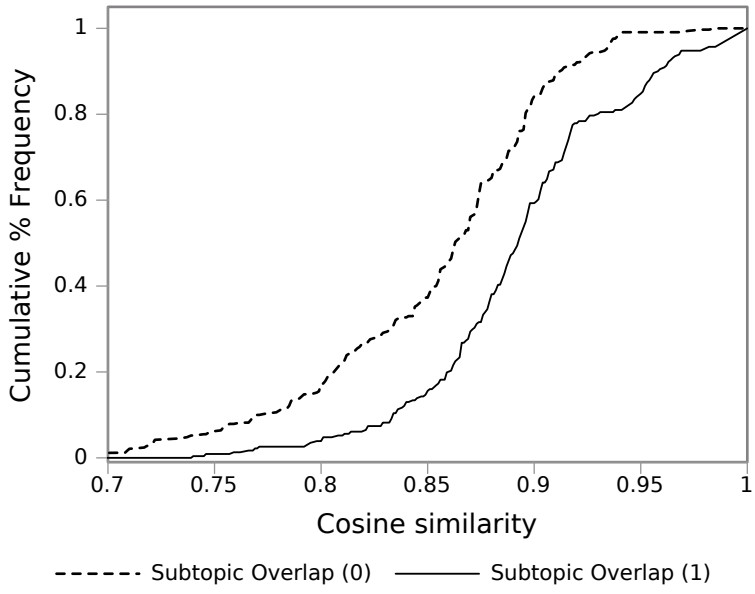


(b) Topic 40

Figure 3.3: Distribution of cosine similarity values for topics 36 and 40.



(a) Topic 44



(b) Topic 49

Figure 3.4: Distribution of cosine similarity values for topics 44 and 49.

As we noted earlier, subtopic overlap provides a crude measure of document similarity. Since current evaluation measures for novelty and diversity essentially measure similarity in this crude fashion, it may be possible to develop new measures of novelty and diversity-aware evaluation measures that incorporate more traditional measures of similarity. Such measures might operate by combining manual assessments of broad topic relevance with automatic assessments of specific inter-document similarity, avoiding the need for explicit subtopics. This work provides a first step in that direction.

To make this speculation a little more concrete, consider a ranked list of documents $\langle d_1, d_2, ... \rangle$. Let Z_k be the set of relevant documents above rank k. Let $sim(d_k, Z_k)$ be an appropriate measure of the similarity between a relevant document at rank k and the set of relevant documents above it. We might then replace the cascade gain value in a typical novelty measure (Clarke et al., 2011) with

$$1 - f(sim(d_k, Z_k)),$$

where the function f serves to convert the similarity value into an appropriate loss value. This idea is discussed in more detail in Section 3.4.

3.3 Evaluating inter-document similarity measures

Inter-document (document to document) similarity measures are required for document clustering and classification, and the cosine similarity remains standard for these applications. While enormous effort has gone into evaluating query-to-document similarity measures, an area which has now advanced far beyond cosine similarity, relatively little effort has gone into evaluating measures that provide inter-document similarity. Noting that subtopics from the diversity task of the TREC Web Track can provide a crude measure of inter-document similarity based on human assessments of relevance, we evaluate several standard similarity measures. Surprisingly, frequency-based similarity measures and measures that are based on data compression ratios substantially outperform the cosine similarity and several other simple measures.

In this section, we discuss approaches that are used to evaluate inter-document similarity measures. We conclude the section by describing our proposal to use subtopic judgments as the basis for evaluating measures of inter-document similarity.

3.3.1 Introduction

The cluster hypothesis has long been the fundamental basis for evaluating inter-document similarity measures in information retrieval. Some research efforts that have attempted to investigate the validity of the cluster hypothesis in document collections include the work of Jardine and van Rijsbergen (1971), Voorhees (1985), Tombros and van Rijsbergen (2001), and Smucker and Allan (2009). Given a query, the *cluster hypothesis test* provided by Jardine and van Rijsbergen (1971) compares distributions of distances between relevant–to–relevant document pairs and non-relevant–to–relevant document pairs. If there is a significant separation between the two distributions, they claimed the document collection satisfies the cluster hypothesis.

Voorhees (1985) criticized the cluster hypothesis test as being sub-optimal since the frequency of non-relevant–to–relevant document pairs is usually significantly more than that of relevant–to–relevant document pairs. Hence, the relative frequency of non-relevant–to–relevant document pairs is significantly less than the relative frequency of relevant–to–relevant document pairs. She also pointed out that the ideal test of the cluster hypothesis should be determined by the number of non-relevant documents that are significantly similar to relevant documents. Her n nearest neighbor (*n*NN) test was designed to test for the number of non-relevant documents that are significantly similar to relevant documents. The n nearest neighbor method is built on the notion that if the cluster hypothesis holds for a corpus, then the n nearest neighbors of a relevant document should also be relevant documents. The n nearest neighbors of a relevant document are the n most similar documents to the particular document. They utilized inter-document (cosine coefficient) similarity between a document and the n nearest neighbor documents at higher ranks (their n was arbitrarily set to 5) for evaluating the cluster hypothesis.

Tombros and van Rijsbergen (2001) proposed query-sensitive similarity measures (QSSM) as a better alternative category of document similarity measures for evaluating the cluster hypothesis in document collections. They claimed the similarity between documents is not independent, but dependent on specific contexts which are usually represented as queries. Apart from the static (absolute and non-changing) similarity between two documents, Tombros and van Rijsbergen postulate that inter-document similarity should incorporate dynamic similarity. The dynamic similarity they proposed is required to be between document pairs and a context which is provided by the given query. Given two documents in a given corpus, the similarity

between the documents will vary depending on a given context represented by queries. Using the cosine similarity measure as the baseline and 5NN as effectiveness measure to evaluate the cluster hypothesis, they claimed their QSSM significantly outperforms the baseline measures in most cases.

Smucker and Allan (2009) investigated the n nearest neighbor method more carefully and concluded that using the n nearest neighbor method alone for evaluating the cluster hypothesis is insufficient for query-sensitive similarity measures in which case document similarity is biased with a given query. They argued that Voorhees' 5NN evaluation method includes documents that only cluster locally, but ignores the global associations and connectivity between documents. They explained that if the nNN test is solely utilized, it is possible to have relevant documents that locally cluster because the cluster contains relevant documents that are very similar to the source document used as a query, but less similar to other relevant documents. They compared the local document association method of Voorhees with a global document association which they called normalized mean reciprocal distance (nMRD) between all relevant documents. In order to compute nMRD, all relevant documents are represented in a relevant document network which is a graph whose nodes are relevant documents and edges represent a notion of relatedness between document pairs. Their algorithm takes as input a measure of relatedness between a relevant document D and all other documents relevant to D. They utilized the rank of a relevant document in a retrieval result of which D is the query as the relatedness measure for their method. Based on their result, neither a local nor a global test measure is sufficient by itself, but a combination of both will reveal different and interesting properties of the cluster hypothesis from the document collection.

Aslam and Frost (2003) utilized TREC's dataset and relevance judgments as a framework for computing similarity values that are comparable across various similarity measures. When a document is judged relevant to a query, they treat the document as a query. They also treat the similarity measure they are evaluating as a retrieval function which they use to perform document retrieval task on the corpus. Retrieved documents are evaluated using the standard mean average precision (MAP). The MAP score for each similarity measure in their consideration is compared.

We borrow their idea of using standard evaluation framework as a means to evaluate the effectiveness of various similarity measures. Following the work of Aslam and Frost (2003), we make use of relevant documents only. We do not perform document retrieval from the corpus

all over again. This is because we can achieve their purpose for performing retrieval using subtopic judgments that provide both the set of relevant documents as well as an indication of their similarity that is based upon the common subtopics they share. We take subtopic judgments as an indication of similarity between documents. Unlike their approach that perform retrieval on the entire collection, we require only the set of relevant documents and their subtopic judgments. Since we already know all the judged relevant documents in the collection, and we have a good indication of their similarity based on the common subtopics they share, we can skip the retrieval aspect of their method.

Test collections developed for the 2009-2011 TREC Web Track decompose each query topic into a set of *subtopics*, with each subtopic providing a different perspective on relevance. Documents are judged independently with respect to each subtopic. Given two documents, each judged relevant to a subset of the subtopics, we define the *overlap* between the documents as the size of the intersection of the subsets. While overlap provides only a crude measure of document similarity, it is a measure that is based on human judgment. Thus, it can form the basis for evaluating other similarity measures. Next, we provide a detailed description of our method.

3.3.2 Method

We have previously shown that documents having common subtopics also have higher inter-document similarity values (Akinyemi and Clarke, 2011). The diversity task of the TREC Web Track requires explicit subtopic categorization of relevant documents. When documents are judged relevant to the same subtopic, there exists a notion of similarity between the documents. Hence, subtopic overlap serves as a "crude measure" of inter-document similarity. We investigate the effectiveness of standard inter-document similarity measures using subtopic overlap as the ground truth. We explore various similarity measures in order to evaluate how well they correlate with the ground truth.

Our goal is to compare the effectiveness of various similarity measures that may be used for inter-document similarity. We consider similarity measures whose scores are bounded between 0 and 1 that also satisfy the *metrics* properties (Tversky, 1977). If $d(x, y)$ represents the distance between two objects x and y, the metrics properties which include: self-similarity $d(x, x) = d(y, y)$, zero property $d(x, x) = 0$, positive property $d(x, y) \geq 0$, symmetry $d(x, y) = d(y, x)$,

and minimality axiom $d(x, y) \geq d(x, x)$ have been previously described in Section 2.6. We select the cosine similarity measure to represent the Vector Space Model's (Salton et al., 1975) feature representation paradigm because it is the most widely-used similarity measure. Two measures – Dice coefficient (Dice, 1945) and Jaccard index (Real and Vargas, 1996) – were selected to represent frequency-based feature representation approach, while mutual information (specifically pointwise mutual information) represents the information-theoretic (Lin, 1998; Aslam and Frost, 2003) approach.

Data compression ratio (Cilibrasi and Vitányi, 2005) is another metric that has been used for estimating the similarity between objects. We also consider the data compression ratio (*CR*) method as a similarity metric. We implemented a very simple data compression ratio that represents the bit rate required to encode a document to another one. For a pair of documents, we obtain the rate of encoding each of the documents to the common terms shared by both documents. A combination of the two ratios (described in Section 2.4.5) becomes a similarity measure.

In sum, we evaluate the effectiveness of eight measures, i.e. cosine similarity, Dice coefficient, Jaccard index, pointwise mutual information, and four combinations of the data compression ratio method.

In the TREC Web Track, human assessors provide relevance judgments for documents relevant to each of the queries. These queries are either ambiguous or under-specified. Therefore, various aspects and interpretations of the queries are identified. These aspects and interpretations are generally referred to as the *subtopics* of a query. In addition to document relevance judgment, each relevant document is further categorized into the subtopics they are relevant to. By applying the diversity-oriented cluster hypothesis (*closely associated documents tend to be relevant to the same interpretations and aspects of an information request*) on the TREC Web Track relevant documents, it is possible to obtain a very reasonable classification of inter-document similarity. Documents that are relevant to the same subtopic should be more similar than those that share no common subtopic.

For each of the queries, we pair all the relevant documents. We also count the number of common subtopics between each document pair such that if documents d_i and d_j share two common subtopics, their subtopic overlap is 2. Thereafter, using the eight similarity measures previously

discussed, we compute inter-document similarity scores between each pair of documents. On each subtopic overlap value, we obtain average inter-document similarity score.

3.3.3 Experimental Details

We used data from the diversity task of the 2009 and 2010 TREC Web Tracks. We removed stopwords and apply a very basic stemming[5] to the documents. Next, we present our result and provide some discussions on the result.

3.3.4 Result and Discussions

Figures 3.5 to 3.12 contain plots showing the performance of the eight inter-document similarity measures in our study at various subtopic overlap values. Each plot shows the average inter-document similarity values obtained at various subtopic overlap values.

The number of documents having subtopic overlaps 4 and 5 in 2009 and 2010 respectively are very low. The number of documents sharing four common subtopics in 2009 is 0.04%. Similarly, the number of documents sharing five common subtopics in 2010 is 0.01%. We omit this low-frequency occurrences from our plots. In all cases, the similarity measures grow linearly as documents share more common subtopics. Figure 3.13 contains a summarized plot of all the similarity metrics on a single plot for the 2009 result. Figure 3.14 shows the summary of the plots for the 2010 result.

Unlike the frequency-based and compression ratio-based similarity measures, the difference in cosine similarity scores between document pairs having no common subtopic and those having at least one common subtopic is very minimal. This indicates that cosine similarity did not distinctly distinguish between documents that are very similar and those that are either not similar or at best marginally similar. Surprisingly, results obtained for mutual information is inconsistent. The particular reason for this inconsistent behavior is unknown, but it has been observed that

[5]For the stemming, if a token ends with *sses*, it gets replaced with *ss*; if a token ends with any of *ies*, *ied*, or *y*, it gets replaced with *i*; if it ends with *tion* or *tions*, it gets replaced with *te*; and if it ends with *ed*, *ing*, *ness*, or *s* and not *ss*, it get replaced with a blank character.

mutual information tend to favor low-frequency tokens. A detailed investigation will be required to detect the specific reason for the mutual information's behavior.

All the frequency-based measures as well as data compression ratio measures provide reasonable results. Similarity score at subtopic overlap K is consistently much higher than the score at subtopic overlap $(K-1)$ for both the frequency-based and data compression ratio-based measures. In fact, there is a steep linear growth of similarity scores from lower subtopic overlap values to higher values. This indicates that frequency-based and data compression ratio-based similarity metrics satisfy our criteria for evaluating the quality of inter-document similarity measures.

3.3.5 Summary

We have presented a method for evaluating the effectiveness of measures of inter-document similarity. Our method utilized subtopic judgments of relevant documents performed by human editorial judges. To the best of our knowledge, our work is the first attempt at utilizing subtopic judgments for evaluating the effectiveness of inter-document similarity measures.

In our study, inter-document similarity measures that are based on either data compression ratio or term occurrence frequency consistently outperform cosine similarity and pointwise mutual information in both datasets we considered. The method may be explored further in order to derive a comparable score for the measures of inter-document similarity in consideration. We leave this as a future work.

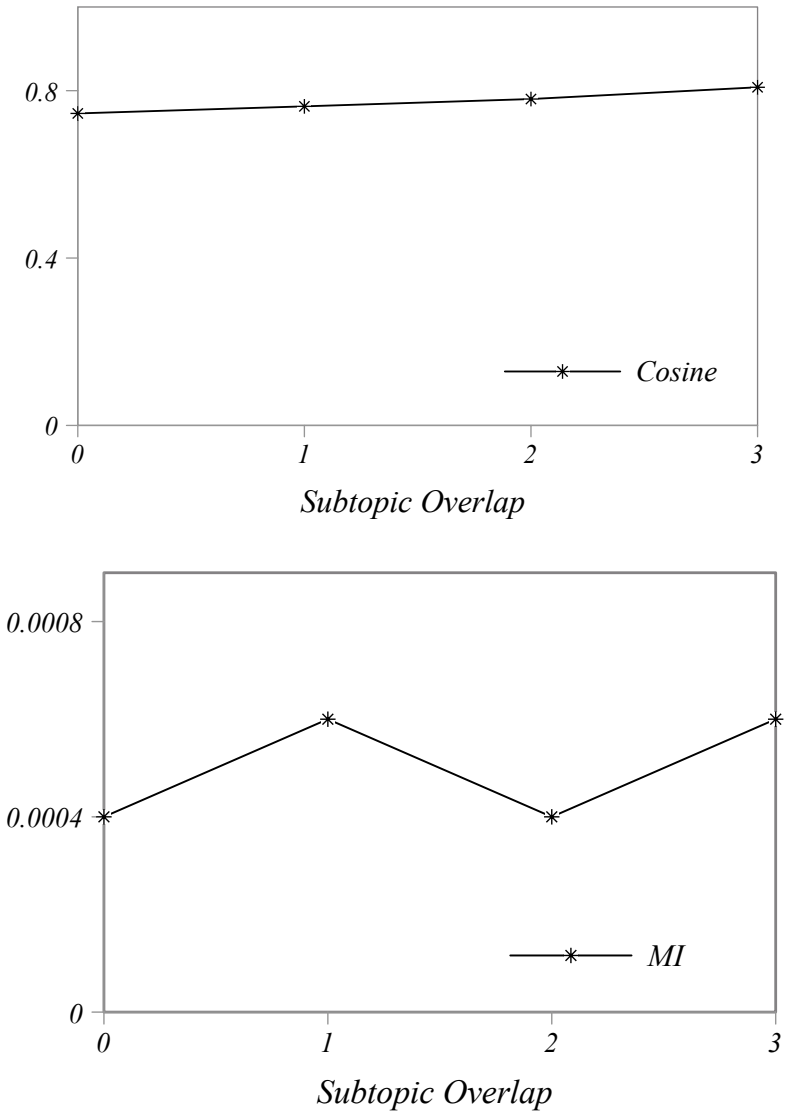


Figure 3.5: Similarity Metrics on TREC 2009: Cosine and Mutual information.

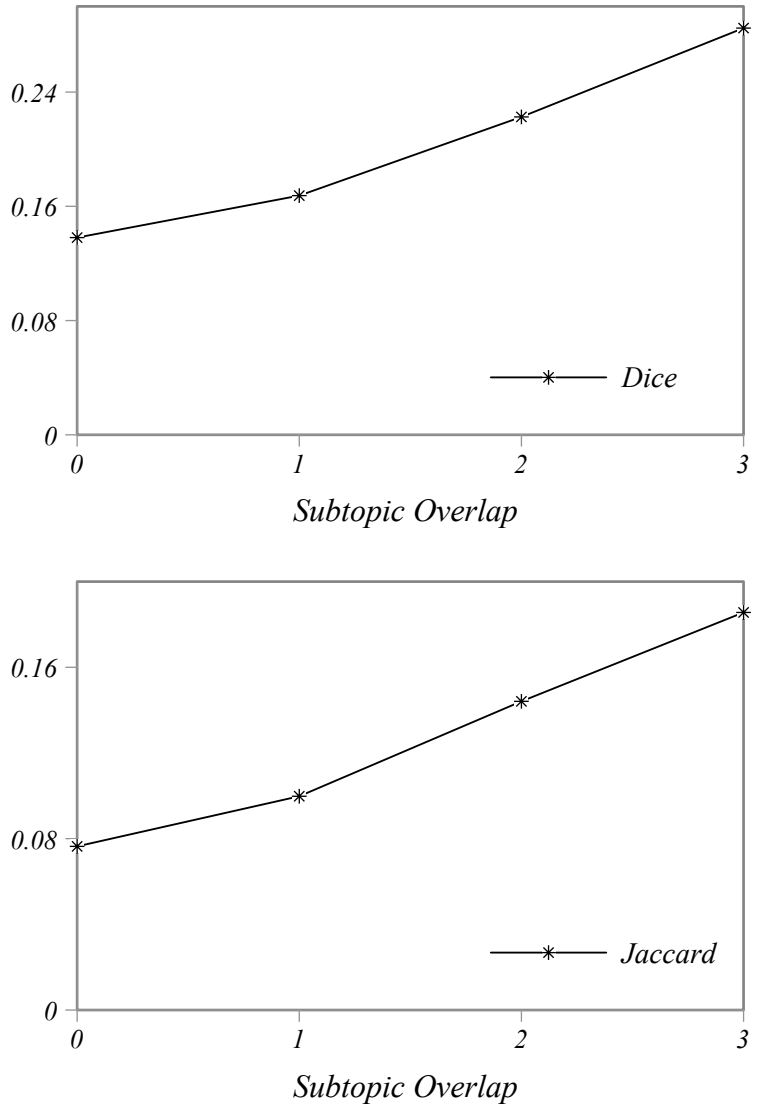


Figure 3.6: Similarity Metrics on TREC 2009: Dice and Jaccard.

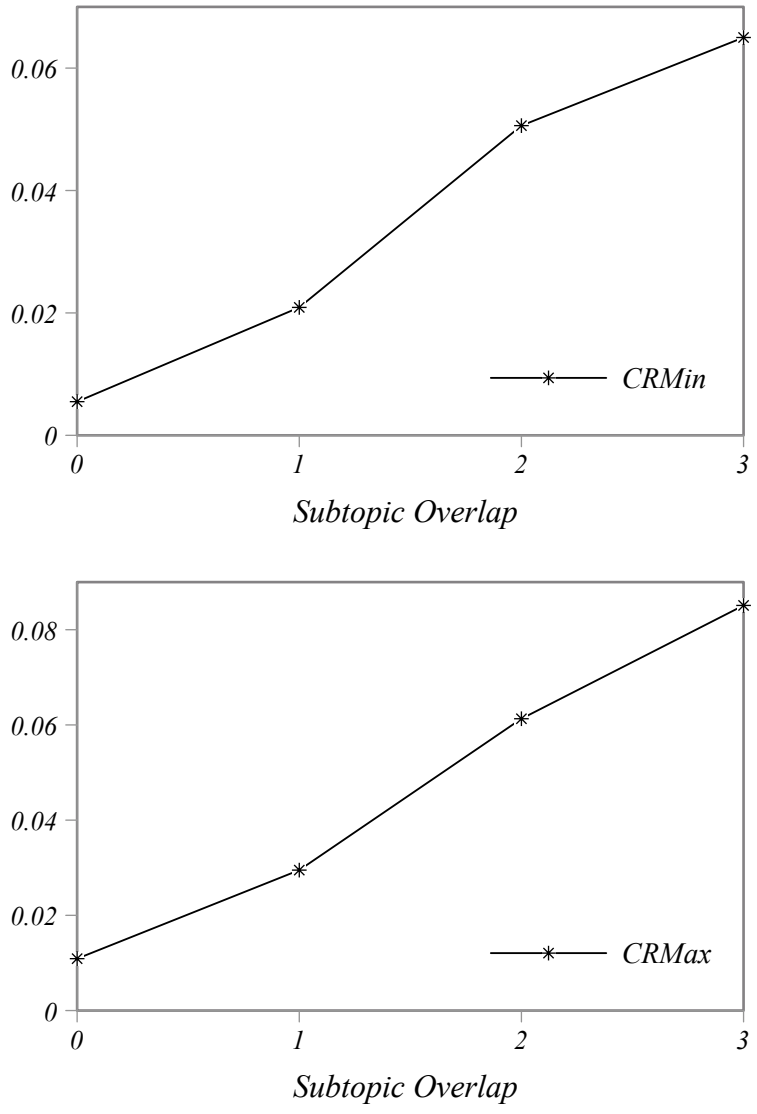


Figure 3.7: Similarity Metrics on TREC 2009: Compression ratio (with minimum and maximum ratio).

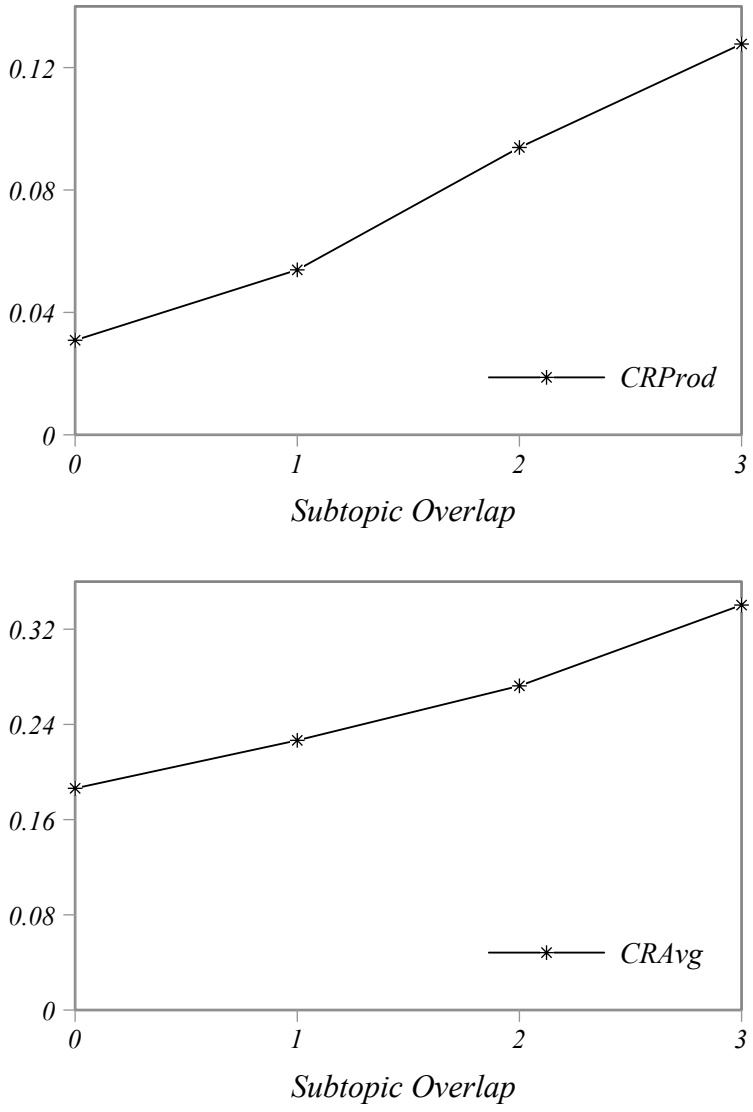


Figure 3.8: Similarity Metrics on TREC 2009: Compression ratio (with product and average).

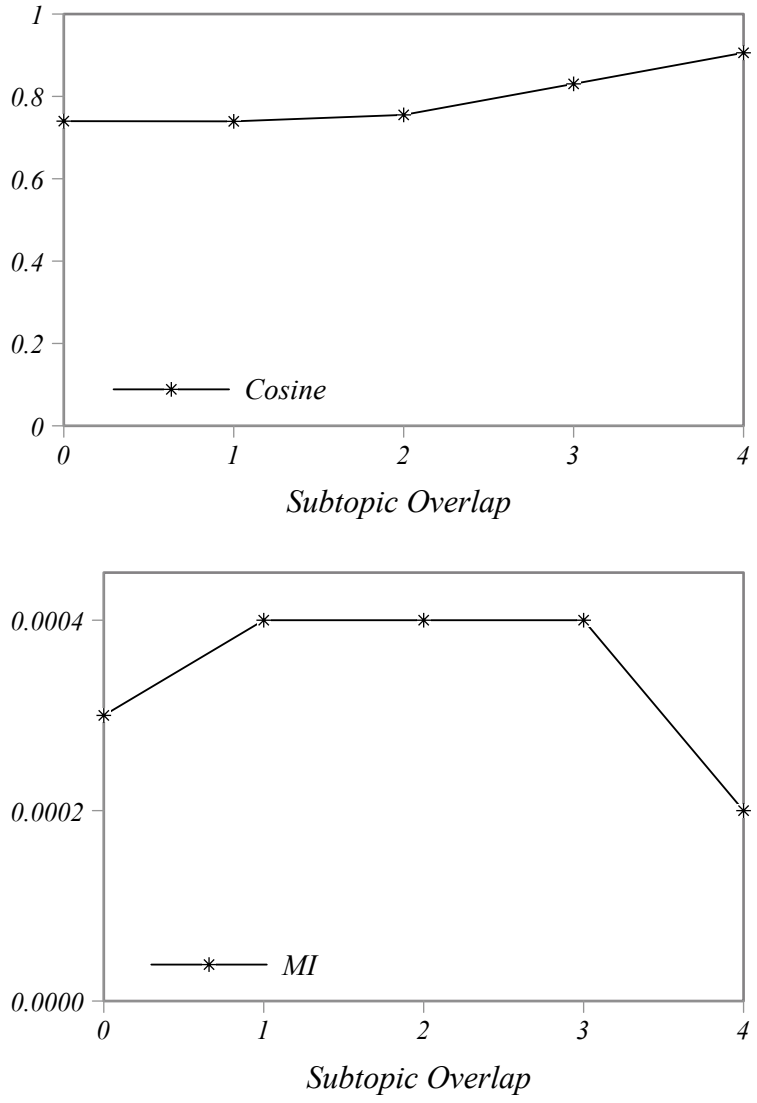


Figure 3.9: Similarity Metrics on TREC 2010: Cosine and Mutual information.

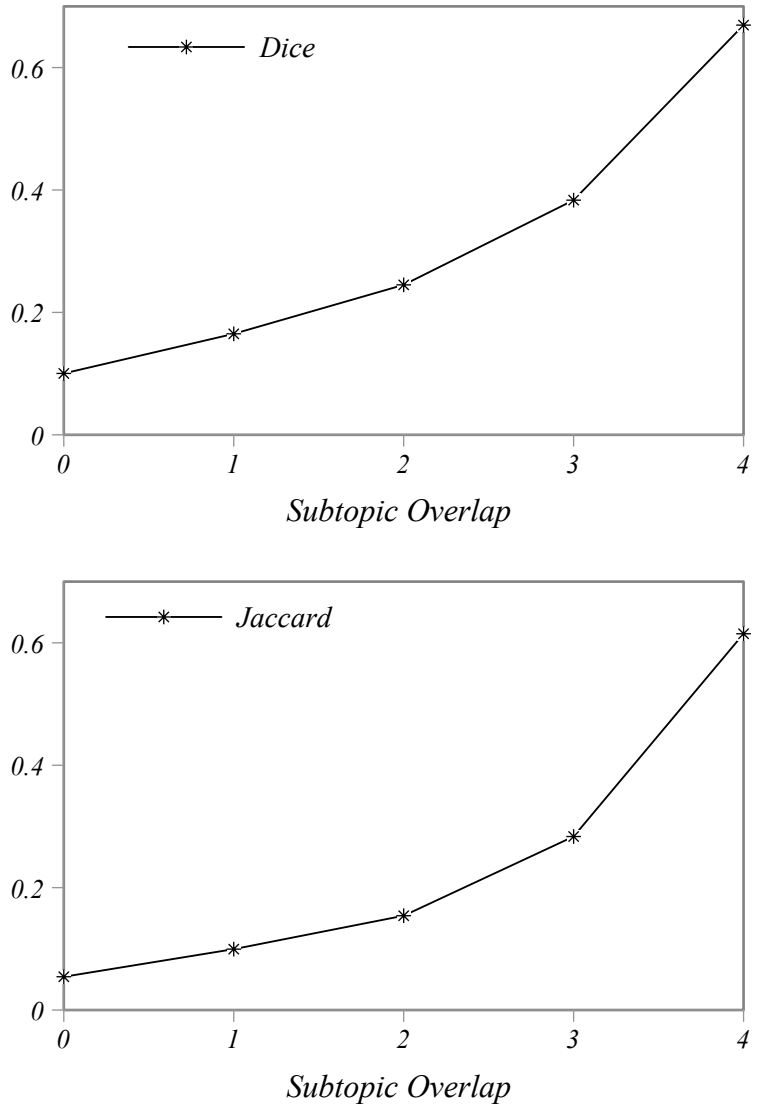


Figure 3.10: Similarity Metrics TREC 2010: Dice and Jaccard.

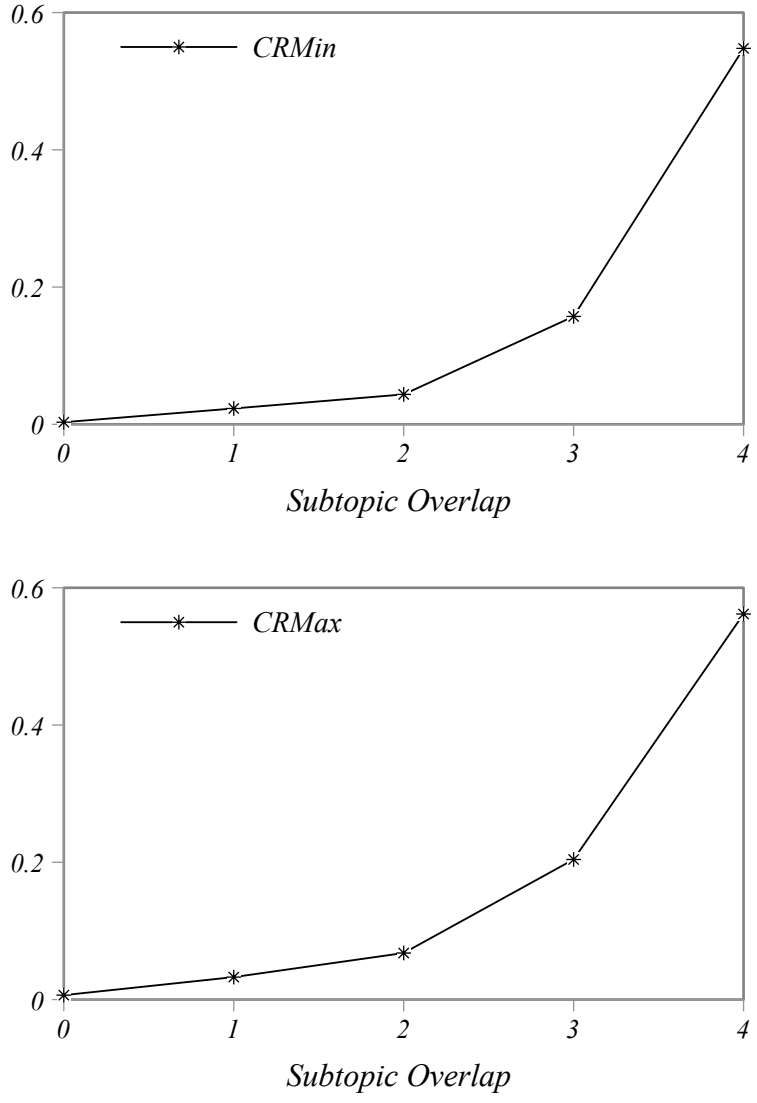


Figure 3.11: Similarity Metrics on TREC 2010: Compression ratio (with minimum and maximum ratio).

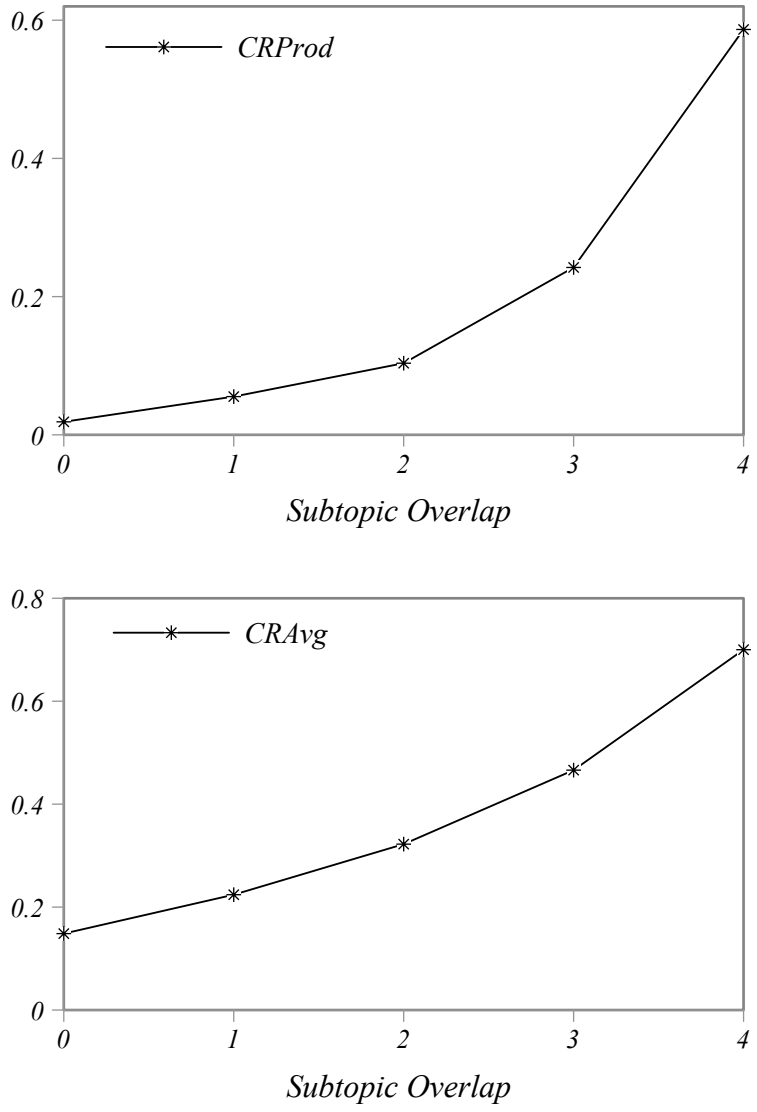


Figure 3.12: Similarity Metrics on TREC 2010: Compression ratio (with product and average).

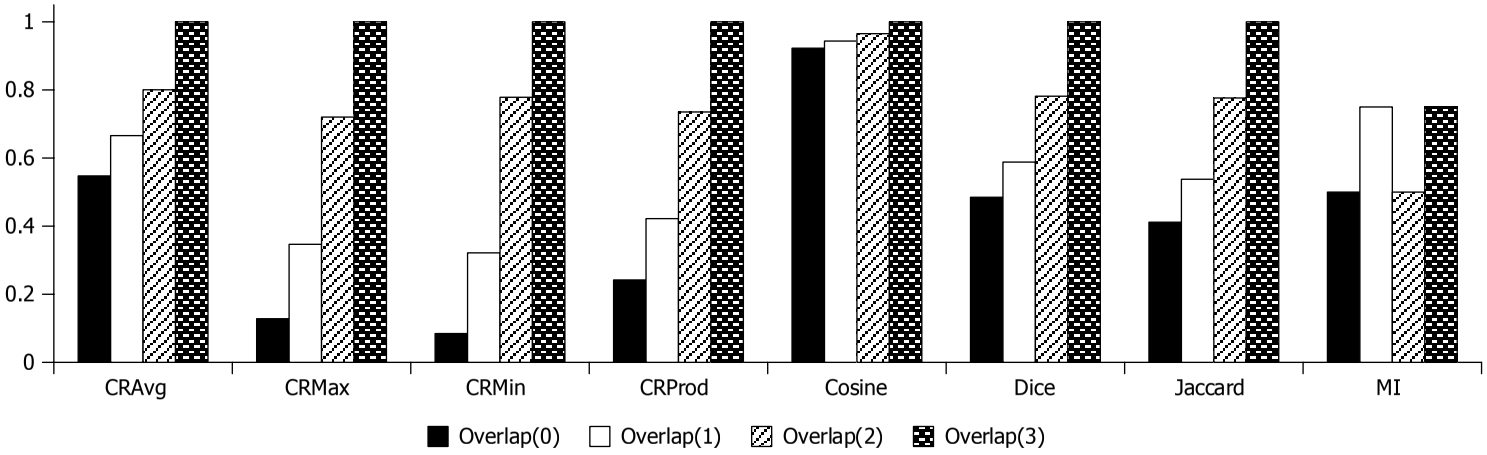


Figure 3.13: Similarity metrics on TREC 2009 dataset: all metrics.

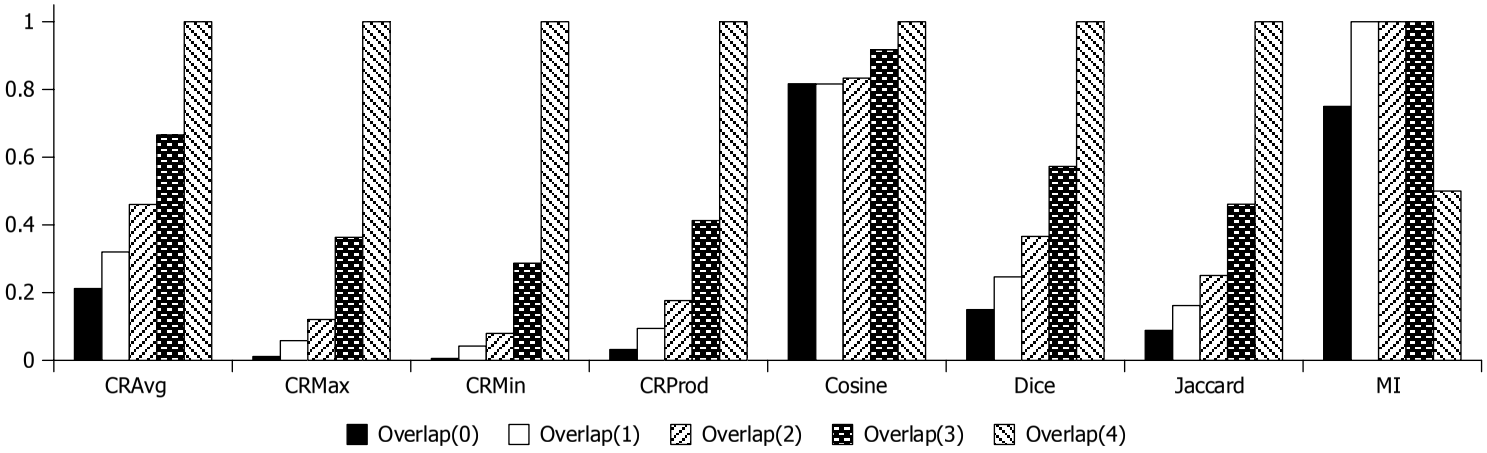


Figure 3.14: Similarity metrics on TREC 2010 dataset: all metrics.

3.4 Diversity evaluation: inter-document similarity method

In this section, we provide a model for evaluating the effectiveness of diversity–aware retrieval functions using inter-document similarity as a substitute for explicit subtopics incorporated into current measures for evaluating the effectiveness of diversity–aware retrieval functions.

3.4.1 Introduction

Evaluation measures for novelty and diversity in information retrieval attempt to reward document relevance and information newness or novelty while simultaneously eliminating redundant information in the result. So far, the state of the art evaluation measures for novelty and diversity–aware ranking models, such as α-ndcg, $ERR\text{-}IA$, and $D\sharp$ all require explicit subtopic judgment of relevant documents.

We have previously shown a positive correlation between subtopic judgments and inter-document similarity. We hypothesize that inter-document similarity measure is a suitable replacement for explicit subtopic judgment. In this section, we investigate our hypothesis both theoretically and experimentally. In particular, we select the cascade models of diversity evaluation, using the Expected Reciprocal Rank (ERR) (Chapelle et al., 2009) as a representative and we explore whether there is a positive correlation between intent aware version of ERR and ERR that incorporates inter-document similarity measures.

Intent-aware versions of the ERR (ERR-IA) is an evaluation model built on top of the ERR model. The ERR-IA has been adopted by the organizers of the TREC Web track for evaluating the effectiveness of diversity–aware retrieval models in information retrieval tasks. The model mimics the *cascaded* behavior of search users by judging the relevance of a document with respect to other documents already seen (i.e., documents at higher ranks.) It extends the reciprocal ranking model by introducing graded relevance to cater for situations when some relevant documents in a ranked result list are more relevant to the user's information need than others. Relevance of a document is also measured with respect to how much information satisfaction a user already enjoyed from higher ranked documents.

Chapelle et al. (2009) discussed user behavioral models in search and contrasted the position-based and cascade-based user models. Position-based models are anchored on modeling the

probability that the document at a particular rank will satisfy the user information need. This model largely ignores whether the user's information need has been satisfied by other higher-ranked relevant documents. Cascade models on the other hand considers the positional relevance with respect to all relevant documents at higher ranks, i.e. previously seen relevant documents. In the cascade model, a user is assumed to scan the document list from top to lower ranks. At each rank, a probability score, denoting the user's satisfaction is assigned to the document at the rank. All documents that are yet to be seen are discarded as soon as the user's information need is satisfied, irrespective of the relevance and position of these documents. As more documents provide additional information satisfaction to the user search need, the overall satisfaction probability increases.

Each relevant document contributes a certain amount of information gain G. G_r is the gain function denoting the information gain a user enjoys for looking at the document at rank r. It is computed using Equation 3.1.

$$G_r = R_r \prod_{i=1}^{r-1}(1 - R_i). \tag{3.1}$$

$$ERR = \frac{1}{D}\sum_{r=1}^{n} G_r. \tag{3.2}$$

R_r is a probability score representing the relevance of a relevant document at rank r out of all n documents in the result list. The ERR for a ranked document list is computed with Equation 3.2 where D is a normalization factor. The user satisfaction or relevance probability score R_r is calculated according to Equations 3.3 and 3.4 where g_r is the graded relevance of the document at rank r, g_{max} is the maximum graded relevance value, and f is a function that maps graded relevance to their probability of relevance.

$$R_r = f(g_r). \tag{3.3}$$

$$f(g_r) = \frac{2^{g_r} - 1}{2^{g_{max}}}, g_r \in \{0, .., g_{max}\}. \tag{3.4}$$

Chapelle et al. (2009) and Clarke et al. (2011) discussed several graded relevance models suitable for the cascaded ERR evaluation models. In cascade models, the relevance of a document at rank r is discounted with respect to the relevance of other relevant documents ranked above r.

Equation 3.2 is modified to cover diversity-oriented evaluation. This is referred to as the intent aware ERR, i.e., ERR-IA. Equation 3.5 shows the information gain for each subtopic t at each successive ranks r, which is denoted by G_r^t. G_r^t denotes the gain obtained for subtopic t at rank r. ERR-IA is computed according to Equation 3.6 where t represents a subtopic of a given query q. The probability $P(t|q)$ denotes a score representing the importance of subtopic t in the query.

$$G_r^t = R_r^t \prod_{i=1}^{r-1}(1 - R_i^t). \tag{3.5}$$

$$ERR\text{-}IA = \sum_{r=1}^{n} \frac{1}{D} \sum_{t} P(t|q) G_r^t. \tag{3.6}$$

ERR-IA is a straightforward evaluation model. However, the real challenge is that the process of obtaining subtopic judgment is quite expensive. It is also a subjective and consequently error-prone process.

We have discussed the positive correlation between subtopic overlap and inter-document similarity in Section 3.2. Our goal is to investigate the utility of inter-document similarity measure as a replacement for the subtopic gain probability $P(t|q)$ which is also the subtopic component included in the ERR-IA model. We intend to replace the subtopic judgment that rely on editorial judgments performed by assessors with measured inter-document similarity–based relevance. We refer to our model as the ERR with inter-document similarity (ERR-IDS).

3.4.2 ERR–IDS Method

We consider a ranked list of n documents $\{d_1, d_2, .., d_n\}$. If Z_r represents the set of relevant documents above the rth ranked document and $sim(d_r, Z_r)$ represents a measure of similarity between a relevant document at rank r and the set of all other relevant documents ranked above

r, i.e. $\{d_1, .., d_{r-1}\}$. Equations 3.7 and 3.8 show a gain computation that is based upon inter-document similarity rather than the intent-aware approaches previously discussed.

$$P(d_r|Z_r) = f(1 - sim(d_r, Z_r)). \tag{3.7}$$

$$ERR\text{-}IDS = \frac{1}{D}\sum_{r=1}^{n} P(d_r|Z_r) \cdot G_r, \tag{3.8}$$

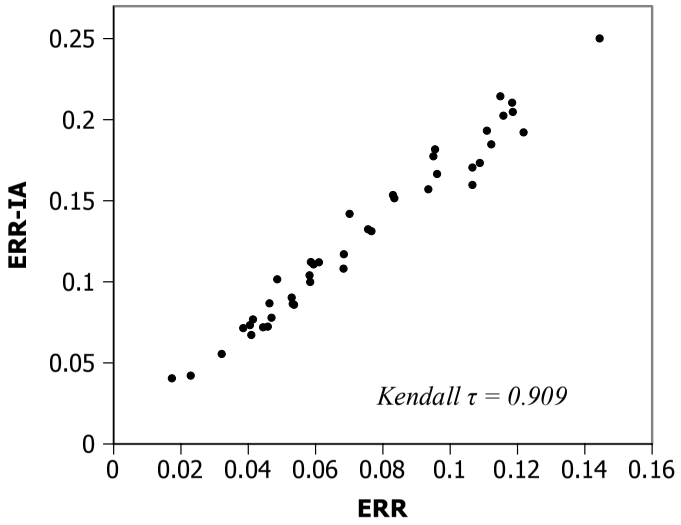
where $P(d_r|Z_r)$ is the probability estimate that d_r contains a new (i.e. novel) information. f is a function that converts inter-document similarity into a probability estimate $P(d_r|Z_r)$ that the document at rank r (d_r) contains new and relevant information not in all previously seen documents at higher ranks. The definition of the function f that converts inter-document similarity measure into a probability estimate is still a work-in-progress requiring further work. G_r — which is the gain a user enjoys for looking at the document d_r — is computed according to Equation 3.1.

In our proof of concept experimentation, we set $P(d_r|Z_r)$ for the first relevant document to the maximum 1.0. This is because Z_r is still an empty set at this rank, and the similarity between the first relevant document and an empty set is zero, i.e., $sim(d_1{}^6, Z_r) = 0$ and consequently $P(d_r|Z_r) = 1.0$. We implemented a very simple method for combining all documents in Z_r by setting Z_r to be the concatenation of all relevant documents that are ranked higher than r. Next, we present an evaluation of our ERR-IDS method.

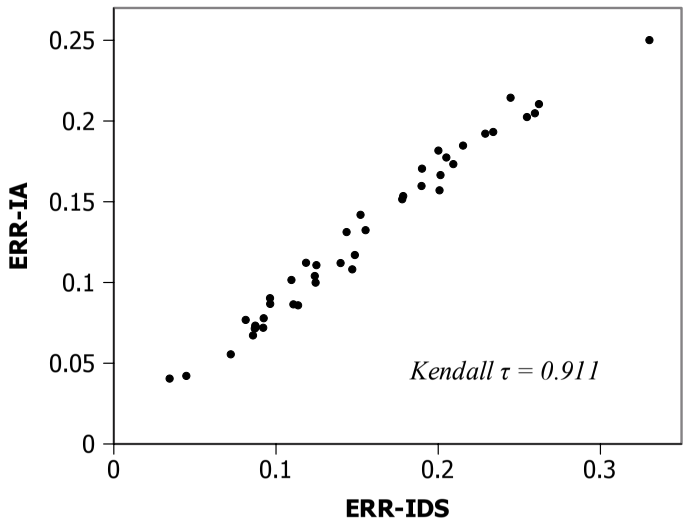
3.4.3 Evaluation of ERR-IDS Method

In order to evaluate the effectiveness of our ERR-IDS method, we take the official ERR-IA evaluation score provided by the TREC Web Track organizers as our ground truth. We compared the correlation of our method with ERR and ERR-IA scores for the diversity task of the Web Track in 2009 and 2010. We select one similarity measure, i.e., CRProd from the list of inter-document similarity measures discussed in Section 2.4.5 and plug its values into ERR-IDS. All the measures are at rank $r = 20$.

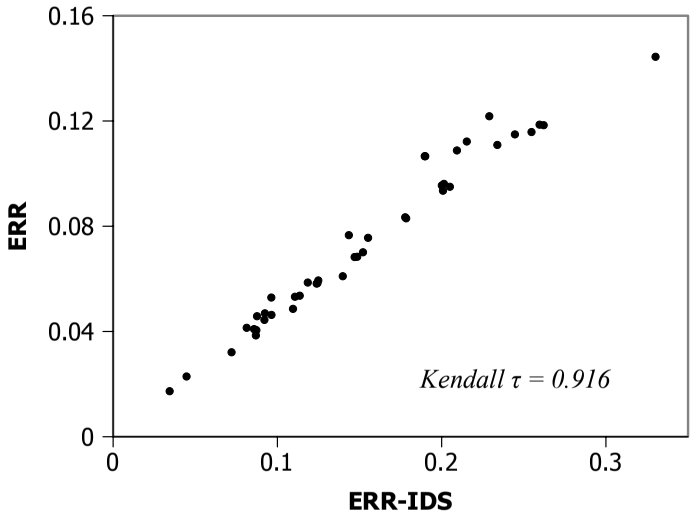
[6] 1 refers to the first relevant document



(a) ERR vs. ERR-IA.

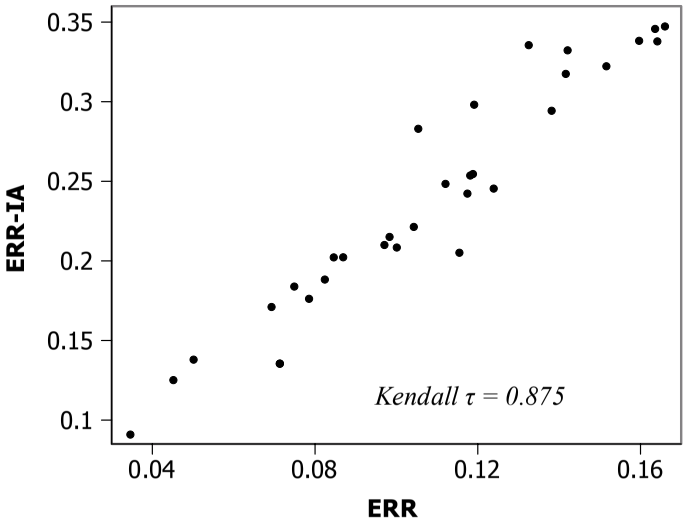


(b) ERR-IDS vs. ERR-IA.

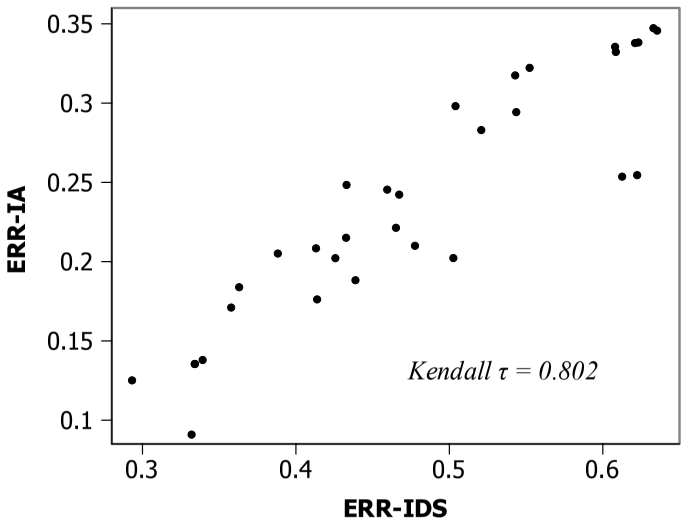


(c) ERR-IDS vs. ERR.

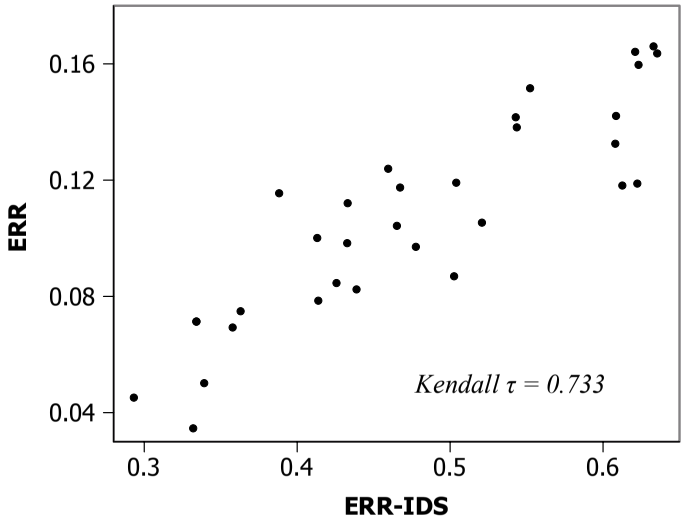
Figure 3.15: TREC Web Track: 2009.



(a) ERR vs. ERR-IA.



(b) ERR-IDS vs. ERR-IA.



(c) ERR-IDS vs. ERR.

Figure 3.16: TREC Web Track: 2010.

TREC Year	ERR-IDS vs. ERR-IA	ERR vs. ERR-IA	ERR-IDS vs. ERR
2009	0.911	0.909	0.916
2010	0.802	0.875	0.733

Table 3.1: Kendall's Tau coefficient: TREC Web (diversity) Track 2009 and 2010.

3.4.4 Discussion

Figures 3.15 and 3.16 contain plots showing the correlation between ERR-IDS, ERR-IA, and ERR in the TREC 2009 and 2010 datasets respectively. The correlation between ERR and ERR-IA shown in Figures 3.15(a) and 3.16(a) are taken as the baseline for 2009 and 2010 datasets respectively. In the 2009 dataset, Figure 3.15(b) shows the correlation between our ERR-IDS and ERR-IA. Likewise, Figure 3.16(b) shows the correlation between ERR-IDS and ERR-IA in the 2010 dataset. For completeness purpose, Figures 3.15(c) and 3.16(c) are also presented. As shown, there is a positive correlation between ERR-IA and ERR-IDS inter-document similarity measure. Using the Kendall's τ coefficient, we measured the rank correlation between ERR-IA and ERR-IDS simultaneously using the correlation between ERR-IA and ERR as a baseline. Table 3.1 shows values of the correlation.

The correlation between ERR-IDS and ERR-IA in 2009 (0.911) is slightly higher than the correlation between ERR and ERR-IA (0.909). We also show the correlation between ERR-IDS and ERR (0.916). The correlation scores are very similar. Likewise, in the 2010 dataset, the correlation between ERR-IDS and ERR-IA is 0.802. Between ERR and ERR-IA, the correlation value is 0.875, and between ERR-IDS and ERR, it is 0.733. In both datasets, the correlation between ERR-IDS and ERR-IA is similar to the correlation between ERR and ERR-IA. This positive and similar correlation between the ERR-based measures is an indication that our method that uses measured inter-document similarity performs comparably with ERR-IA that requires explicit subtopics in its model. We believe the positive and similar correlation between the measures is an indication that our method should be investigated further on additional datasets and additional inter-document similarity measures. Even though we are also of the opinion that our method may subsequently mature into replacing subtopic-based evaluation models for diversity, additional extensive work is still required to finalize the ERR-IDS evaluation model. Our current effort provides a step in that direction.

3.4.5 Summary

We have investigated the utility of inter-document similarity for evaluating the effectiveness of diversity–aware retrieval functions. We presented some theoretical basis of our evaluation

method. To the best of our knowledge, our work is the first attempt at utilizing measured inter-document similarity as an effectiveness measure for diversity–aware retrieval models. We believe this method warrants further exploration in order to gain more understanding of the relationship between diversity and inter-document similarity. In the next chapter, we present our work in the area of subtopic mining and intent discovery.

Chapter 4

Intent Discovery

This chapter is organized into two parts. The first part in Section 4.1 describes the notion of query intent – its purpose and the approaches used to obtain it. Sections 4.2 and 4.3 in the second part present two approaches we have explored for discovering query intent in this thesis. As a result of our goal to explore alternative approaches that are independent of user interaction data for uncovering query intent, none of our two approaches make use of user interaction logs, query logs, and other organized data sources. We uncovered diverse query intents directly from the corpus. Our first approach utilizes pseudo-relevance feedback obtained from top k retrieval results. The second approach employs anchor text and anchor links to uncover diverse query intents.

4.1 Query Intent

Search involves an active interaction between two major entities: A *user* presumably having an information need, i.e. *user intention*, and a text collection providing the user information need as a ranked search result. User information need is usually represented as queries in information retrieval tasks. As a result of short queries supplied by search users, user information need provided to the search engine are often ambiguous and under-specified. For example, various users might provide the query "*windows*" to a search engine. Some of the users might be satisfied with infor-

mation about Microsoft Windows operating systems or other aspects of the Microsoft software company. Others might be interested in replacement windows only, X Windows, the Windows musical group, or the Windows movie. Topically relevant documents for each legitimate user need are mostly mutually exclusive. The challenge facing information retrieval systems is how to satisfy the information need for all or most search users having varied query intentions as represented by their ambiguous or under-specified query "*windows*".

A basic first step is to understand when a query is ambiguous. An understanding of when a query is ambiguous is crucial for retrieval systems in order to know whether query disambiguation is required for a query or not. In the same token, if a query is under-specified, there might be need for the search engine to perform an automatic query expansion for the query. Another very important information is to have a knowledge about various user intentions. When most user intentions for an ambiguous query is known, the top k result may be modeled to satisfy either all or most known non-trivial intents of the query. Otherwise, the information need of some search users might not be satisfied in the top k result.

Predicting diverse query intents for an ambiguous query is difficult because the same search engine serves several users with varied intentions. In most cases, a single interpretation for an ambiguous query will not satisfy all users. Different users have different interpretations of a query. A solution provided in the literature is the explicit diversification of search results in order to cover various important query intents (Carbonell and Goldstein, 1998; Akinyemi et al., 2010; Clarke et al., 2008; Santos et al., 2010b; Radlinski et al., 2010; Carterette and Chandar, 2009). A search engine could retrieve documents that satisfy various query intents and hope most users will be satisfied by having their information need within the top k of retrieval result.

Currently, search engine query logs provide a primary source of data for query intent discovery (Radlinski et al., 2010; Song et al., 2009). Web query and user interaction logs may be mined to support both ranking and evaluation models that reward diversity and punish redundancy (Clarke et al., 2009; Radlinski et al., 2010; Song et al., 2009). In order to diversify user queries, user interaction, query log and click information are used for tuning Web search engines. This method has been reported to be an effective approach for diversifying results. By considering previous users and learning their search behaviors and intent choices, subsequent search results can be diversified using the learned behaviors of previous users. This way, search results are skewed to match the intentions of search users learned from previous user behaviors. This

approach is particularly suitable for Web search engines because of the huge size of data they have access to and the inherent diversity of their users.

The NTCIR-9 Intent Task organizers provided a framework for evaluating query intent discovery algorithms. Query intent discovery is expressed as a subtopic mining problem. Research groups were provided queries that were either ambiguous or under-specified. They also provide a corpus of Chinese documents downloaded from the Web as well as the corresponding query log for the task. They encouraged task participants to submit mined subtopics which are phrases representing diverse possible intents of the original query. Available approaches for mining subtopics from queries are summarized in the overview paper for the task (Song et al., 2011). Generally, the approaches used by participating research groups are a combination of either one or more of the following methods: query log mining, information from commercial search engines, encyclopedias, and information from anchor text and anchor links.

In the next two sections, we present the two approaches we have explored for uncovering possible query intents or mining subtopics of an ambiguous or under-specified query. In the next section, we describe our content-oriented method in which case we uncover diverse query intents from the top k search results using a method similar to pseudo-relevance feedback of important terms in the top k result. The subsequent section presents a description of our approach founded on anchor text and anchor link information.

4.2 Intent Discovery: Pseudo-relevance feedback

In this section, we present our method that uses pseudo-relevance feedback of terms in the top k retrieval result as a means to directly uncover diverse subtopics of a given query from the corpus. The discovered subtopics represent diversified query intents.

4.2.1 Introduction

Pseudo-relevance feedback (Ruthven and Lalmas, 2003) has been used successfully in information retrieval to improve the quality of search results.

Given a query consisting of terms, the idea is to identify terms from the top k result that are non-trivially related to the terms in the given query. Term co-occurrence and term proximity are two of the methods used in practice to select these related terms. The terms are extracted and further analyzed in order to improve the quality of an initial ranked result. Extracted terms may be used to expand an original query or they can provide terms suitable for query recommendations. The query expansion process should be done very carefully in order to avoid the expanded query from drifting to satisfy another information need that is different from those initially intended.

We borrow the pseudo-relevance feedback idea and utilized it in the area of query intent discovery. Our idea is to take as input terms from the top k documents that are most related to the given query term and compute a term relatedness score between these terms. Related terms may be clustered based on their score that represents the extent of their relatedness in the corpus as a form of similarity or distance measure between the terms. Distinct clusters of related terms are considered as distinct query intents.

4.2.2 Method

We begin by taking as input a query and performing retrieval on the corpus. Out of the retrieved top k documents and using a term scoring function, we select terms that are related to the given query. These terms are clustered into groups according to their measure of relatedness obtained using a similarity measure. The clusters are ranked, from which diverse query intents are subsequently selected. Next, we provide a detailed description of our approach.

Retrieval

Several approaches such as document or passage retrieval may be utilized as a retrieval function for the initial document retrieval. If a standard document retrieval function is used, document terms may be scored using a term-weighting function such as tf-idf. On the other hand, passage retrieval may also be used. Passages are short compared to full sized documents. Passage retrieval takes into consideration term occurrence, term proximity, and term frequency when computing scores for passages.

We opted to implement a passage retrieval function because the retrieved passages are short. This eliminates a lot of non-related terms and prevents them from being included in subsequent post-retrieval processing. We retrieve m passages that are relevant to the original query. The value of m necessarily needs to be large enough in order to uncover enough important and diverse passages. We arbitrarily set m to 200.

Term Pruning and Clustering

Our intention is to group terms having high relatedness scores together in a cluster. Terms from retrieved passages are weighted and ranked according to their co-occurrence frequencies with the original query. Terms that co-occur with the original query more often are selected. We arbitrarily select 200 terms. These terms are clustered using their pointwise mutual information (pmi) as the clustering criteria.

Considering two terms t_i and t_j with individual document frequencies $|t_i|$ and $|t_j|$ in the corpus and $|t_i \cap t_j|$ representing the frequency of the documents they co-occur in. Their pmi score pmi(t_i, t_j) is computed as:

$$\text{pmi}(t_i, t_j) = log \frac{p(t_i, t_j)}{p(t_i) \cdot p(t_j)}, \tag{4.1}$$

where $p(t_i)$ and $p(t_j)$ are the respective probabilities of the discrete events of independent occurrences of t_i and t_j in the corpus; $p(t_i, t_j)$ is the probability of both t_i and t_j co-occurring in the corpus. We utilized their maximum likelihood estimate as their probability values, such that,

$$p(t_i) = \frac{|t_i|}{|t_C|}, \tag{4.2}$$

$$p(t_j) = \frac{|t_j|}{|t_C|}, \text{and} \tag{4.3}$$

$$p(t_i, t_j) = \frac{|t_i \cap t_j|}{|t_C|}, \tag{4.4}$$

where $|t_C|$ is the total number of documents in the corpus. We compute the pmi score for pairs of terms. Using this score, we group the passages into clusters of similar or related passages. There are many clustering algorithms that could be used. For this work, we used the Girvan-Newmann (Girvan and Newman, 2002) clustering algorithm because of its suitability to handle graphs having a large number of nodes.

Cluster Ranking

Obviously, some of the clusters are more related to the original query than others. Therefore, there is need to rank the clusters in order to show the extent of their relatedness to the query. First, we rank the terms in each cluster according to their measure of relatedness to the query. After which the clusters are also ranked according to their relatedness to the query. Terms in each cluster may be ranked based on their co-occurrence frequencies with the original query. If a term t_i co-occurs with the original query 1000 times, and another term t_j co-occurs with the query 50 times, t_i will be ranked higher than t_j.

In order to rank the clusters, the highest ranked term for each cluster is selected to represent each cluster. The co-occurrence frequencies of the highest ranked terms are also used to rank the clusters. The resulting ranked clusters represent distinct and diverse intents of the given query.

4.2.3 Evaluation

Since there is no standard reusable evaluation framework we can use to directly compare the effectiveness of our method, we performed both an informal and a non-standard evaluation on the generated clusters of diverse query intents or mined subtopics. We utilized the 2004 TREC Novelty track collection which uses the AQUAINT[1] dataset. The AQUAINT corpus contains documents extracted from the New York Times, the Associated Press, and the Xinhua News Agency newswires. It consists about one million news articles. The articles in the New York Times and the Associated Press were created between 1999 and 2000, while those in the Xinhua News Agency were created between 1996 and 2000.

[1] http://www.ldc.upenn.edu/

The Novelty track organizers provided fifty topics which can potentially retrieve novel documents from the dataset. Tables 4.1 and 4.2 show two representative clusters for topics N58 and N60. Topic N58 is about *irradiated food safety* while topic N60 is about the *abortion pill RU-486*. In the tables, the numbers in the bracket written in front of each term indicate the number of documents that contains both the original query and the term. These numbers are used to rank the terms in each cluster as well as the entire clusters.

Informal Evaluation: Topic N58 — "irradiated food safety"

Most clusters in topic N58 shown in Table 4.1 are self explanatory. However, the seventh cluster which is "*california (17), harmful (15), opposed (8), law (8)*" seems not obviously related to the query. But, further analysis using a popular search engine reveals there is a California law that requires that food companies warn consumers of products that contain chemicals that can cause reproductive harm or cancer. A legal case in 2004 between California State Attorney General Bill Lockyer vs. Starkist, Chicken of the Sea, and Bumble Bee contested the requirement that food makers issue such warnings on food labels. Therefore, the cluster is relevant to the query and it actually provides a novel information.

Similarly, the ninth cluster, i.e. *hearing (10), davis (10), smith (8), caroline (5)* also seems not obviously related to the query. But, in 2004, Caroline Smith DeWaal was the director of food safety for the Center for Science in the Public Interest (CSPI) in the U.S.A. On March 30, 2004, she testified at the U.S. House Committee on Government Reform Subcommittee on Civil Service and Agency Organization Washington, D.C. with a speech titled "A System Rued: Inspecting Food" which discussed the subject of irradiated food safety. Again, the cluster is relevant to the query and it also provides another novel information.

Informal Evaluation: Topic N60 — "abortion pill RU-486"

Another topic, N60 shown in Table 4.2, is about the abortion pill RU-486. The first cluster covers the approval of the pill, its relationship with a federation (National Abortion Federation) and the access to the pill. The second cluster covers the relationship of the pill with women, for a plan (family planning), and with "France and French" (the pill was initially available in France before

other countries.) The third cluster covers the identification of the pill as a politically controversial drug for pregnancy-related purposes. The fourth cluster covers the administration, cause, and control of what the pill might be used for. The fifth cluster identifies a need to be conscious of the effectiveness of the pill. The sixth cluster covers the marketing and safety aspect of the pill. The seventh cluster covers antagonism against the pill, and cluster number eight identified that the pill is associated with a choice to either leave or terminate an event (pregnancy). Each of these clusters provide novel information.

Cluster	Terms
1.	consumer (34), products (31), reduce (23)
2.	bacteria (30), meat (29), eating (26)
3.	processing (29), federal (22), testing (20)
4.	healthy (29), sick (11), outbreaks (10)
5.	produce (28), university (21), research (19)
6.	kill (22), require (20), handle (16), regulations (9)
7.	california (17), harmful (15), opposed (8), law (8)
8.	chemical (13), nuclear (8), engineered (8), radioactive (7)
9.	hearing (10), davis (10), smith (8), caroline (5)

Table 4.1: Topic N58: Irradiated Food Safety

Cluster	Terms
1.	approved (128), federation (84), access (42)
2.	women (119), plans (105), french (88), france (88)
3.	drug (117), politics (91), pregnancy (74)
4.	administration (103), cause (58), control (48)
5.	effective (76), consider (38)
6.	provide (72), market (64), safe (58)
7.	anti (63), school (33), associations (26), students (25)
8.	leave (24), terminate (22)

Table 4.2: Topic N60: Abortion Pill RU-486

Non-Standard Evaluation

As an additional evaluation of our method, we carried out a non-standard evaluation of the clusters. In order to measure the quality of the clusters, we performed the evaluation by combining query expansion and document retrieval operations. Our goal is to utilize inter-document similarity measures as an evaluation method for our clusters. We selected the five highest scoring clusters and used their terms as query expansion terms on the original query. We performed a document retrieval using the original query for the top k relevant documents setting k to 20. We used this document set as the *baseline* result.

Using the query expansion terms, we also performed document retrieval for the top k relevant documents. Twenty relevant documents from each group of clusters is selected in a round-robin manner. We label the document set as the *expansion* result.

We hypothesize that the average inter-document similarity score of the *baseline* run will be higher than that of the *expansion* run. We are of the opinion that the difference in inter-document similarity scores is a consequence of the novelty introduced into the result set through our method. We note that it is also possible that the documents retrieved using expanded queries might have drifted away from being topically relevant to the original query.

As a consequence of using terms that are very related to the original query, we hypothesize that the result set obtained from query expansion will still be very relevant to the original query. At the same time, duplicates and some near duplicate documents would be avoided. In order to test our hypothesis, we measured the divergence between the original query and the two sets of results from both the *baseline* and *expansion* methods.

In summary, we obtain relevant documents for each query before query expansion, i.e. *baseline*. After which we obtained a measure of their inter-document similarity. Next, we expand the queries with terms in the clusters and use expanded queries for retrieval. The average inter-document similarity scores between pairs of documents in the *baseline* experiment is predicted to be higher than the average inter-document similarity score between document pairs in the *expansion* experiment. After expansion, we obtain another set of relevant documents and measured their inter-document similarity. The divergence of the document list to the original query is measured using KL divergence. We predict that the inter-document similarity score for the *baseline* run will be higher than those from the *expansion* run. We also predict that the divergence of the

baseline run result list to the original query will be lower than the divergence of the *expansion* run result list to the original query. Hence, we performed a paired t-test on the distributions. We obtained both one-tailed and 2-tailed p-values for the two distributions.

Table 4.3 shows the result for both the inter-document similarity as well as document divergence to the original query. We implemented seven inter-document similarity measures discussed in previous chapters. The result shows they all support our inter-document similarity hypothesis. The p-values obtained from all the distributions show statistical significance between the inter-document similarity scores. This indicates that these inter-document similarity methods detected that the *baseline* result list tend to have documents that are more similar and consequently less diverse.

Our hypothesis on divergence is also supported by the KL divergence scores obtained and shown in the table. The *expansion* run result list has a higher divergence from the original query than the *baseline* run result list. However, this divergence is not statistically significant with respect to the p-values of 0.064 and 0.128 obtained for one-tailed p-value and two-tailed p-value respectively. Both p-values are higher than the 0.05 threshold for statistically significant phenomenon.

4.2.4 Summary

We have demonstrated an alternative approach for uncovering query intents from document corpus. Our method utilized pseudo-relevance feedback as a means to discover terms (directly from a document corpus) that are related to the terms in a given query. Values obtained from co-occurrence frequencies of terms are used to prune all the terms. The remaining related terms are clustered using their pointwise mutual information as a distance measure between term pairs. Our result indicates the method is promising especially since the document corpus is generally available for researchers unlike the current state-of-the-art that utilizes query logs which are not generally available. In the next section, we describe our subtopic mining method that utilized anchor text and anchor link information.

	Baseline	**Expansion**	% Δ	p-Value (1-tail)	p-Value (2-tails)
CRMin	0.054	0.044	18.42	1.58×10^{-4}	3.15×10^{-4}
CRMax	0.062	0.052	17.17	1.11×10^{-4}	2.21×10^{-4}
CRProd	0.097	0.083	14.26	8.80×10^{-6}	1.76×10^{-5}
Dice	0.242	0.223	8.21	1.70×10^{-6}	3.42×10^{-6}
Jaccard	0.237	0.220	7.34	9.30×10^{-6}	1.86×10^{-5}
CRAvg	0.266	0.247	7.25	1.36×10^{-5}	2.71×10^{-5}
Cosine	0.791	0.762	3.71	1.00×10^{-6}	1.98×10^{-6}
KL Divergence	0.323	0.330	-2.26	0.064	0.128

Table 4.3: Mean inter-document similarity score: TREC 2004 Novelty Track.

4.3 Intent Discovery: Anchor text

In this section, we present our method that uses anchor text, out-links of anchor text, and the target documents of anchor text to uncover diverse subtopics from text corpora. The subtopics that are uncovered represent diversified query intents. Anchor text and their target documents are represented as nodes on an *anchor text-target document* graph of which edges are the out-links between an anchor text and its corresponding target documents. Terms that are related to a given query are retrieved from the graph. The terms are eventually clustered into non-overlapping groups denoting distinct aspects of the given query.

We evaluated the utility of our anchor text method on two datasets — SogouT[2] and ClueWeb09[3] — and three evaluation tasks, i.e., 2011 NTCIR-9 Subtopic Mining Task[4], 2009 and 2010 TREC Web Tracks[5]. We utilized queries provided for the topics in each of the evaluation tasks.

4.3.1 Introduction

We explore the utility of anchor text, their target documents, and their out-links to target documents for uncovering query intents and subtopics in Web documents. Two characteristics of anchor text were exploited for uncovering diversified query intents. First, when an anchor text, such as "*windows*" have out-links to various non-related documents such as the Microsoft Windows operating system and replacement windows, one might assume the anchor text is related to diverse subtopics. Queries that are related to an anchor text that contains diverse subtopics are considered either ambiguous or under-specified. Most importantly, the quantity of out-links from an anchor text to various documents containing non-related topical relevance may indicate the extent of diverse subtopics in both the query and anchor text.

The second characteristic applies to the possibility of the same target document being referenced by different anchor text. For example, a document about Microsoft Windows operating

[2] `http://www.sogou.com/labs/dl/t.html`
[3] `http://boston.lti.cs.cmu.edu/clueweb09/`
[4] `www.thuir.org/intent/ntcir9`
[5] `plg.uwaterloo.ca/~trecweb`

system might be the target document out-linked by the following anchor text "Microsoft", "windows", and "operating system". Our method considers the three anchor texts as being related because they all out-link to the same document. Our intention is to uncover terms satisfying both characteristics such that terms satisfying the first characteristic are uncovered and clustered together based on the second characteristic. In the example, terms "microsoft", "operating", and "system" may be grouped together while "doors" and "replacement" may also form another group representing possible intents of the query "*windows*".

By representing out-links of anchor text as edges of a graph of which nodes consist of both anchor text and their target documents, it is possible to obtain from the graph other terms that are related to the terms in the given query. In this work, we extract anchor text and their target documents from text corpora. Both the anchor text and target documents are represented as nodes on an *anchor text-target document* graph. The out-links between an anchor text and the target document are implemented as edges in the graph. The set of anchor text that are related to a given query is retrieved. The out-links of retrieved anchor text are also obtained. Other anchor text that also out-links the same target documents are obtained.

Figure 4.1 is a pictorial depiction of an *anchor text-target document* graph. A_i's represent anchor text and T_j's represent target documents where i and j are discrete integer variables denoting the index of a particular anchor text or a target document. As an example, if a query is related to anchor text A_3, target documents having out-links from A_3 (i.e., T_1, T_2, and T_3) are obtained. Other anchor text having out-links to target documents T_1, T_2, and T_3 are also obtained. The anchor text A_1, A_2, A_4, A_5, and A_6 are also obtained because they have out-links to one or more already obtained target documents.

Terms in all the anchor text set are obtained and clustered together using a notion of their relatedness. In order to obtain a measure of terms' relatedness, pointwise mutual information (*pmi*) of term pairs was utilized. Term co-occurrence frequency and individual term frequencies become the values plugged into the *pmi* function. The *pmi* information between term pairs is suitable to cluster the terms into various groups of related terms, such that each cluster represents a distinct query intent.

4.3.2 Background

In this section, we explore the task of predicting aspects and subtopics, suitable for providing acceptable coverage of queries, by exploiting anchor text and anchor out-link information drawn from the collection itself. We draw our inspiration from the work of Radlinski et al. (2010). They infer subtopics from user query reformulation and co-click information obtained from the logs of a commercial Web search engine. The essential idea of our approach is to replace query logs with anchor text (Dang and Croft, 2010), and to replace co-click information with anchor text co-occurrence information. While anchor text and link information provide core features for Web search, to the best of our knowledge, no detailed investigation has examined how anchor text can be used to uncover query intents.

Anchor text has long been used as a substitute for queries when appropriate logs are not available. Dang and Croft (2010) have explored the use of anchor text as a substitute for query logs. Their idea has been implemented on the ClueWeb09 collection and the anchor text query log[6] generated has been made publicly available. They compared the effectiveness of using query logs versus anchor text for reformulating queries. They concluded that using anchor text as a simulated query log for query reformulation purposes is as effective as using a query log. Similar to their approach, our work also replaces query logs with anchor text information. On the other hand, the problem we address in this work is different from the problem they investigated. They explored anchor text as a replacement for query logs, we investigate using anchor text for uncovering novel aspects of a query.

In early work, Eiron and McCurley (2003) examine the relationship between queries and anchor text, suggesting that the success of anchor text as a feature in Web search arises in part from its similarity to queries. Along the same lines, Kraft and Zien (2004) demonstrate the value of anchor text for query refinement. We continue this work, exploring the ability of anchor text to expose the diversity underlying queries.

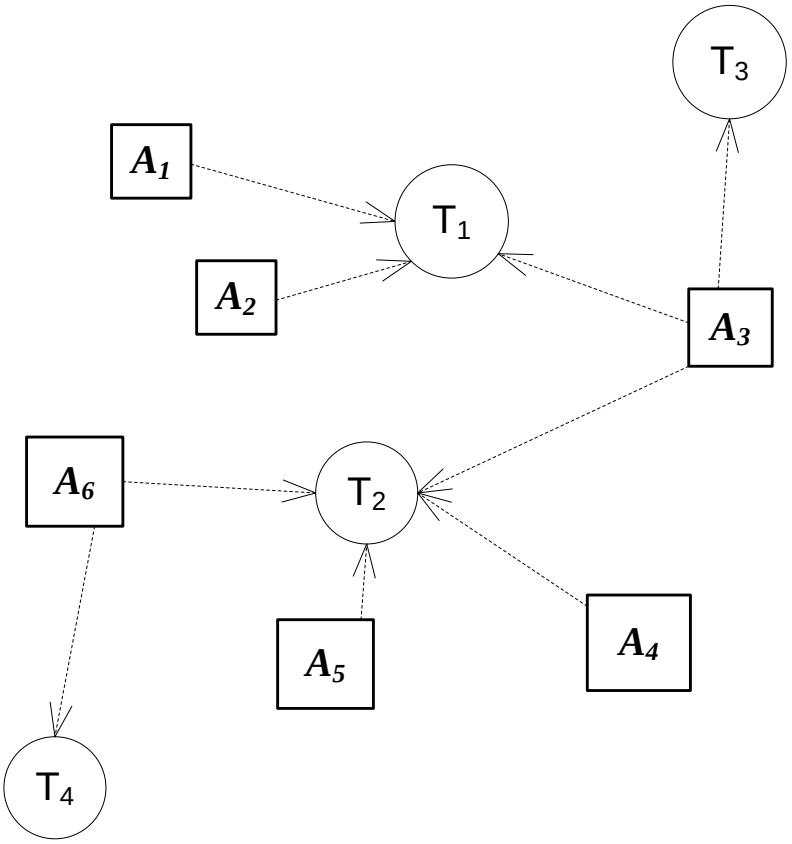


Figure 4.1: Document–anchor text graph.

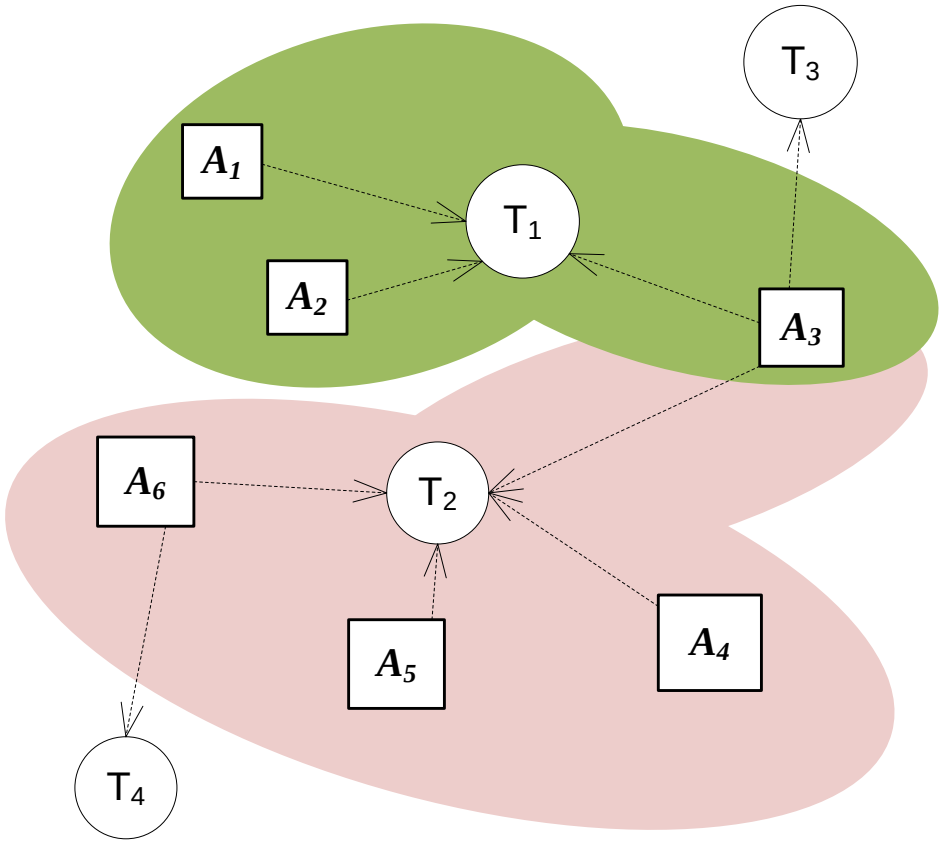


Figure 4.2: Clusters of anchor text on document–anchor text graph.

4.3.3 Method

We start with anchor text extracted from a collection of Web pages (Hiemstra and Hauff, 2010). Anchor text on each page and their corresponding out-links are extracted. Both the anchor text and the document linked to by the out-link are represented as nodes on a document–anchor text graph. The out-links form the edges on the graph and the edges are directed from an anchor text to a target document. Figure 4.1 shows a pictorial description of the document–anchor text graph. In the figure, $A_1, A_2, ..., A_6$ represent six distinct anchor texts, while T_1, T_2, T_3 and T_4 are the target documents of the anchor text.

For each target document, all anchor text associated with the target document are selected based on the out-links referencing the document. From this information, related anchor text to a target document may be obtained by selecting all anchor text referencing a target document. As indicated in Figure 4.2, we see a description of two sets of related anchor text that were generated from Figure 4.1. Two groups of related anchor text may be selected from the graph in Figure 4.2. We have drawn a circle around each group of related anchor text.

The first group of anchor text consists of A_1, A_2 and A_3 while the second anchor text group consists of A_3, A_4, A_5 and A_6. A_3 is an overlapping anchor text that have out-links to both target documents T_1 and T_2. Even though the grouping approach seems reasonable, we encountered a few problems during our experimentation that require further attention. There are more target documents T_is than required. If individual target documents are treated as distinct intent, the number of intents becomes very large and inappropriate for our purpose. We also encountered target documents with singleton linked anchor text. This category does not have enough information redundancy to constitute a distinct intent, especially when there are other related anchor text that are connected to other target documents.

As a result, we select all terms in all of the retrieved anchor text. We compute a relatedness metric between the terms. Underlying this relatedness metric is the assumption that terms may be related if they appear in anchor text linking to a common page, even if they link from different pages. Given a pair of terms t_i and t_j, we compute the relatedness between these terms using pointwise mutual information (pmi):

[6] `http://lemurproject.org/clueweb09/anchortext-querylog/`

$$\text{pmi}(t_i, t_j) = log \frac{p(t_i, t_j)}{p(t_i) \cdot p(t_j)}, \tag{4.5}$$

where $p(t_i, t_j)$ is the probability of t_i and t_j co-occurring in the anchor text associated with a page, while $p(t_i)$ and $p(t_j)$ are the independent probabilities of t_i and t_j occurring in the anchor text of any page. For these probabilities, we use maximum likelihood estimates based on the extracted anchor text.

We extract terms that co-occur with the query from all the anchor text, ranking them by their co-occurrence frequency. We then cluster the terms using the relatedness measure defined above. We also utilized the Girvan-Newmann (Girvan and Newman, 2002) clustering algorithm because of its suitability to handle graphs having a large number of nodes. We view each of the resulting clusters as representing a distinct aspect or subtopic of the original query. Terms from the clusters may subsequently be used as query expansion terms, either to improve novelty and diversity, or to create evaluation subtopics.

We investigated the utility of our approach on two datasets, namely: the SogouT collection of Chinese documents and the ClueWeb09 Category B collection. The SogouT data collection consists of about 130 million web pages crawled and downloaded from the World Wide Web mid-2008 from 5.3 million web sites. The web pages contain documents written in Chinese. It's uncompressed size is 5TB.

The ClueWeb09 dataset was provided by the Language Technologies Institute at Carnegie Mellon University. It was created to support research in information retrieval and related human language technologies. The dataset consists of 1 billion web pages, in ten languages, crawled and downloaded between January and February 2009. The dataset has been adopted for several tracks of the TREC conference. Its size is 5TB when compressed and 25TB uncompressed. The Category B dataset is a subset of the whole Category A dataset. It consists of 50 million web pages written in English language and it is about one tenth of the whole dataset.

We participated in the 2011 NTCIR-9 subtopic mining task. We also implemented our method on the diversity task of the TREC Web Track in 2009 and 2010. Apart from the NTCIR-9 which directly measures query intents, the TREC experiments do not explicitly measure the quality of query intents or mined subtopics. The main focus of the TREC Web track experiments

was the quality of actual retrieved documents. Even if very good quality subtopics are uncovered by an algorithm, if the retrieval system used in conjunction with identified subtopic terms does not retrieve documents that are judged relevant and novel, the evaluation score for the retrieval function will be low. Next, we describe details of our participation at the subtopic mining task of the 2011 NTCIR-9 Intent track.

4.3.4 NTCIR-9 Intent and Subtopic Mining Task

Figure 4.3 shows a pictorial description of the problem addressed in the subtopic mining task of NTCIR-9. The goal of the NTCIR intent and subtopic mining task (Song et al., 2011) is to obtain diverse intents for provided queries from the provided test collection. Given an ambiguous or underspecified query, the information retrieval system is required to uncover non-trivial aspects covering relevant subtopics of the given query. Eight of the one hundred topics provided for the task are shown in Table 4.4.

The task organizers also provided the SogouT collection which consists of Chinese text corpus, a query log (SogouQ) of user interactions associated with the corpus, and queries that task participants are required to provide diverse intents for. The queries were generated with the intention of being either ambiguous or under-specified in order to uncover the underlying intents of each query. We explored query intent discovery using anchor text and anchor link information. When an anchor text that appears in different locations in a source document or various source documents hyperlinks more than one target document, the anchor text is considered to have implicit diverse intents.

Considering "*windows*" (our previously stated example) as an anchor text that hyperlinks various target documents related to the Microsoft company, operating systems, software updates, and replacement windows. These varieties in the target documents indicate the diversified intents that are possibly derivable from the "*windows*" query. We uncovered these implicit intents of the queries using the link information between anchor text and their corresponding target documents.

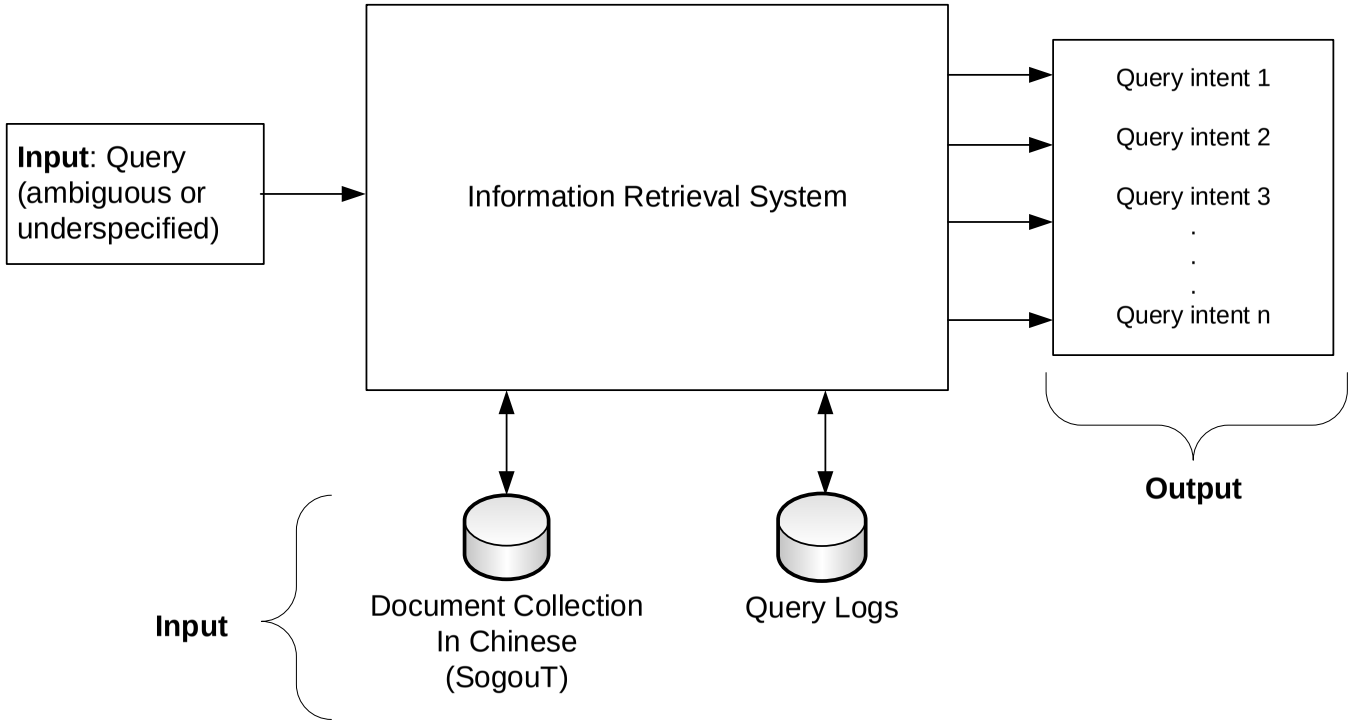


Figure 4.3: NTCIR-9 Intent and Subtopic Mining Task Problem

An example of uncovered subtopics provided for topic 8 is shown in Figure 4.4. The query is written in Chinese, but we include an English language translation of the query as well[7]. Topic 8 is about the *symptoms of diabetes*. We show the first six intents of the nine official subtopics provided for topic 8 by the NTCIR Subtopic mining task organizers in Figure 4.4. They cover subtopics related to:

- the different stages of diabetes symptoms,
- the clinical manifestations of diabetes symptoms,
- causes, prevention and treatment of diabetes symptoms,
- the symptoms of Type II diabetes,
- the juvenile and children diabetes symptoms, and
- gestational diabetes symptoms and diabetes symptoms for pregnant women.

From the provided corpus, we extracted anchor text, the source documents containing them and their target documents. The Chinese characters are encoded into their UTF-8 equivalent. We crudely segmented the anchor text UTF-8 representation into their unigram and bigram tokens. Tuples of ⟨source document, anchor text, target document⟩ were indexed as units of documents.

The provided queries were also segmented into their unigram and bigram UTF-8 equivalents. Using a passage retrieval function on the index with the queries as inputs, we retrieved anchor text and their target documents. For all retrieved target documents that have additional edges of out-links to other anchor text, we further retrieve all the additional anchor text. All the anchor text are ranked and grouped in order to either eliminate duplicates or remove noisy anchor text from the anchor text list. We submitted two runs (`UWat-S-C-1` and `UWat-S-C-2`) to the NTCIR-9 Subtopic mining task for evaluation.

[7]The translation was obtained using the language translation application provided by Google at `translate.google.com`.

```
<topic number="0008">
        <query>糖尿病症状</query>
        <interpretation> Symptoms of diabetes</ interpretation >
        <intent number="1">
                <description>糖尿病症状【不同阶段】</description>
                <interpretation> different stages of diabetic symptoms </ interpretation >
                <examples>糖尿病的早期症状;糖尿病初期症状...</examples>
                <interpretation> Early symptoms of diabetes; the early symptoms of diabetes
                </interpretation >
        </intent>
        <intent number="2">
                <description>糖尿病症状【临床表现】</description>
                <interpretation> Clinical manifestations of diabetic symptoms</ interpretation >
                <examples>糖尿病症状 临床表现;消瘦 头晕 乏力等糖尿病症状...</examples>
                <interpretation> Clinical symptoms of diabetes; weight loss dizziness, fatigue and
                other symptoms of diabetes </ interpretation >
        </intent>
        <intent number="3">
                <description>糖尿病症状【原因、预防、治疗】</description>
                <interpretation> Symptoms [Diabetes causes, prevention, treatment]</ interpretation >
                <examples>糖尿病症状检查诊断;糖尿病症状及原因...</examples>
                <interpretation> Symptoms of diabetes diagnosis; diabetes symptoms and causes
        </interpretation >
        </intent>
        <intent number="4">
                <description>糖尿病症状【种类】</description>
                <interpretation> Symptoms of diabetes [type] </ interpretation >
                <examples>2 型糖尿病症状;二型糖尿病症状...</examples>
                <interpretation> Diabetes symptoms; symptoms of Type II diabetes </ interpretation >
        </intent>
        <intent number="5">
                <description>糖尿病症状【发病人群】</description>
                <interpretation> Onset of diabetes symptoms [people] </ interpretation >
                <examples>儿童糖尿病症状;怎样发现儿童糖尿病患者的症状...</examples>
                <interpretation> Juvenile diabetes symptoms; how to find the symptoms of children with
                diabetes </ interpretation >
        </intent>
      <intent number="6">
                <description>糖尿病症状【孕妇、女性】</description>
                <interpretation> diabetes symptoms [pregnant women]</interpretation>
                <examples>妊娠糖尿病症状;孕妇糖尿病症状...</examples>
                <interpretation>gestational diabetes symptoms; pregnant women, the symptoms of
                diabetes</interpretation>
      </intent>
</topic>
```

Figure 4.4: NTCIR-9 Topic 8 Official Subtopics: subtopics 1 to 5

S/No.	Intent (Chinese)	Interpretation (English)
1.	早期症状糖尿病	early symptoms of diabetes
2.	妇女糖尿病的具体症状	women specific symptoms of diabetes
3.	严重的糖尿病症状	severe diabetes symptoms
4.	糖尿病并发症症状	complications of diabetes symptoms
5.	2 型糖尿病的症状	type 2 diabetes symptoms
6.	糖尿病初期症状和主要症状	the early symptoms and main symptoms of diabetes
7.	糖尿病酮症酸中毒症状	symptoms of diabetic ketoacidosis
8.	抗氧化剂改善糖尿病肾病症状	anti-oxidants to improve the symptoms of diabetic nephropathy
9.	小儿糖尿病的症状与治疗	pediatric diabetes symptoms and treatment
10.	孕期糖尿病症状有哪些？	What are the symptoms of diabetes during pregnancy?
11.	皮肤瘙痒可能是糖尿病症状	skin itching may be symptoms of diabetes
12.	糖尿病患者的心血管系统症状	cardiovascular diabetes symptoms
13.	非酮症高渗性糖尿病昏迷症状及诊断	non-ketotic hyperosmolar diabetic coma symptoms and diagnosis
14.	控制血压可缓解糖尿病肾病症状	control of blood pressure may relieve symptoms of diabetic nephropathy

Figure 4.5: NTCIR-9 subtopics uncovered by our algorithm for Topic 8, i.e. "*symptoms of diabetes*"

Experimental Details

For retrieval, we chose to use an established passage retrieval algorithm, which was originally developed as an initial retrieval step in a question answering system by Clarke and Terra (2004) rather than a traditional document retrieval function because anchor text is short compared to an average document size. Passage retrieval is suitable for retrieval on short documents as well as when the entire document is not required but only a short text fragment is appropriate. Occurrence of query terms as well as their close proximity in an anchor text are incorporated into the scoring function of the passage retrieval algorithm.

Anchor text scoring function

We assign scores to each of the anchor text retrieved by the passage retrieval function. Let $t_1, ..., t_n$ represent an anchor text such that t_1 is the first term in the anchor text and t_n is the last term. Our anchor text scoring function takes as input (i) the total number of query terms q_t and (ii) the ratio of the number of unique query terms q_t^u in an anchor text, and (iii) the total number of terms in the anchor text n. The scoring function is given by the Equation 4.6.

$$score = |q_t| \cdot \frac{q_t^u}{n} \tag{4.6}$$

Using Equation 4.6, we rank all retrieved anchor text. The highest scoring set of anchor text are submitted as our `UWat-S-C-1` run.

Anchor text clustering

Our `UWat-S-C-2` run takes as input the `UWat-S-C-1` run and groups the anchor text that have very strong relationships on the *anchor text-target document* graph. If two or more anchor text out-link edges are connected to a particular target node, we put all the anchor text in the same cluster. Thereafter, the highest scoring anchor text in each cluster is selected to represent the cluster. All the selected clusters are ranked based on their scores and they are submitted as our `UWat-S-C-2` run.

Submissions

Figure 4.5 shows fourteen subtopics uncovered from the collection for topic 8 using our method. Six of the subtopics overlap with subtopics in the official subtopics provided by NTCIR Subtopic mining task organizers. For example, the first and sixth subtopics in our result, which are about *early symptoms of diabetes* correspond to the first subtopic in the official result. The second subtopic in our result which is about *symptoms of diabetes in women* corresponds with the sixth subtopic in the official result. Our fifth subtopic which is about *type 2 diabetes symptoms* corresponds with the fourth subtopic in the official result. The ninth subtopic in our result which is about *symptoms and treatment of pediatric diabetes* corresponds with the fifth subtopic about *symptoms of juvenile diabetes or diabetes in children*. The tenth subtopic in our result which is about the symptoms of diabetes during pregnancy corresponds with the sixth subtopic in the official result about *gestational diabetes symptoms and symptoms of diabetes in pregnant women*.

This example topic shows a high subtopic overlap between our result and the official result provided. As shown in the figure, there are other subtopics discovered by our method which were not provided in the official result. Table 4.5 shows the official evaluation result of our submissions. In all cases, `UWat-S-C-2` outperforms `UWat-S-C-1`.

Summary

We have demonstrated that anchor text usage for subtopic mining is promising. The much better performance of our `UWat-S-C-2` run against the `UWat-S-C-1` run also indicates the utility of anchor text and anchor links as a reasonable criteria for clustering similar anchor text and by extension similar documents. We envisage that a combination of our method and subtopic mining methods that utilize user interaction data extracted from query logs will produce better quality result. We leave this as a future work.

Topic	Query	Interpretation
1	日俄战争	Russo-Japanese War
8	糖尿病症状	Diabetes Symptoms
15	莫扎特	Mozart
33	辅食	food supplement
45	丰田 suv 普拉多	Toyota Prado suv
55	什么是指数基金	What is an Index Fund
85	求职面试技巧	job interview skills
100	中国计划生育政策	Chinese family planning policy

Table 4.4: Sample NTCIR-9 Queries

runid	I-rec			D-nDCG			D#-nDCG		
	@10	@20	@30	@10	@20	@30	@10	@20	@30
UWat-S-C-1	0.239	0.324	0.327	0.249	0.246	0.193	0.244	0.285	0.260
UWat-S-C-2	0.332	0.494	0.511	0.336	0.389	0.315	0.334	0.442	0.413

Table 4.5: Official Evaluation Result

4.3.5 TREC 2009 and 2010 Web tracks

Web query and user interaction logs may be mined to support both ranking and evaluation models that reward diversity and punish redundancy (Clarke et al., 2009; Radlinski et al., 2010; Song et al., 2009). Unfortunately, such logs are not widely available outside the major commercial search engine companies.

We implemented our method that uncovers subtopics and diverse query intents using anchor text, target documents of anchor text, and out-links of anchor text. For queries, we used the queries provided for the TREC 2009 and 2010 Web track. The track included a diversity task in which case selected queries are either ambiguous or under-specified. The track did not directly measure subtopics. To evaluate our work, we performed experiments on anchor text (Hiemstra and Hauff, 2010) extracted from the ClueWeb09 Collection[8]. For queries, we used topics taken from TREC Web Track[9]. These topics are specifically created with diversity experiments in mind (Clarke et al., 2009). Along with a query that might be executed against a search engine, each topic also includes a number of subtopics reflecting different aspects and interpretations of that query. At TREC, these subtopics were used to compute effectiveness measures that explicitly take novelty into account (Clarke et al., 2009). These subtopics were developed from query term clusters created by the method of Radlinski et al. (2010), which depends on the availability of search engine query logs. In part, we aim to replace this method with one requiring only anchor text, and other more readily available information.

Result and Discussion

Tables 4.6 and 4.7 present two typical examples of the output from our approach, both based on topics taken from the TREC 2009 Web track. For these examples we include the top three terms from the top five clusters. Within clusters, terms are ranked by their co-occurrence frequency (given in brackets) with the original query. Clusters are ranked by the most frequent term they contain. The first example in Table 4.6 (for the query "espn sports") shows several major sports appearing in different clusters. The second example in Table 4.7 (for the query "elliptical trainer") shows several popular brands of elliptical trainer.

[8] `boston.lti.cs.cmu.edu/Data/clueweb09`

[9] `plg.uwaterloo.ca/~trecweb`

```
<topic number="15" type="ambiguous">
    <query>espn sports</query>
    <description>
        I'm looking for various sports scores and information from the
        ESPN Sports site.</description>
    <subtopic number="1" type="nav">
            Take me to the ESPN Sports home page.</subtopic>
    <subtopic number="2" type="inf">
            I'm looking for college football and basketball scores.</
    subtopic>
    <subtopic number="3" type="inf">
            I want to find NBA basketball standings.</subtopic>
    <subtopic number="4" type="inf">
        I'm looking for baseball scores and information on upcoming
        live broadcast games.</subtopic>
    <subtopic number="5" type="inf">
            I'm looking for information on NASCAR races.</subtopic>
    <subtopic number="6" type="inf">
            I'm looking for information on fantasy football leagues.</
    subtopic>
</topic>

<topic number="33" type="faceted">
    <query>elliptical trainer</query>
    <description>
            Find information about elliptical trainer machines.</
    description>
    <subtopic number="1" type="inf">
            I'm looking for reviews of elliptical machines.</subtopic>
    <subtopic number="2" type="inf">
            Where can I buy a used or discounted elliptical trainer?</
    subtopic>
    <subtopic number="3" type="inf">
            What are the benefits of an elliptical trainer compared to
    other fitness machines?</subtopic>
    <subtopic number="4" type="inf">
            What are the best elliptical trainers for home use?</
    subtopic>
</topic>
```

Figure 4.6: TREC 2009 Web Track Topics 15 and 33.

Topic 15: espn sports
1. football (430), nfl (294), night (60)
2. games (396), international (306), college (289)
3. basketball (356), schedule (114), district (108)
4. radio (299), hockey (210), league (98)
5. winter (299), magazine (266), action (122)

Table 4.6: Clustering examples for TREC 2009 Web Track topics — Topic 15.

Topic 33: elliptical trainer
1. fitness (859), trainers (685), machine (219)
2. cross (856), machines (228), kettler (88)
3. equipment (459), product (282), shopping (181)
4. proform (180), horizon (105), spirit (70)
5. schwinn (171), stamina (119), track (51)

Table 4.7: Clustering examples for TREC 2009 Web Track topics — Topic 33.

	2009		2010								
	α-nDCG @20	**ERR-IA @20**	**ERR-IA @5**	**ERR-IA @10**	**ERR-IA @20**	**α-nDCG @5**	**α-nDCG @10**	**α-nDCG @20**	**strec @5**	**strec @10**	**strec @20**
baseline run	0.272	0.071	0.206	0.222	0.231	0.249	0.291	0.327	0.299	0.424	0.537
expansion run	0.278	0.075	0.223	0.233	0.240	0.276	0.305	0.336	0.354	0.439	0.539
p-value (2-sided paired t-test)	0.358	***0.035***	***0.044***	0.133	0.158	***0.027***	0.195	0.269	***0.020***	0.343	0.468

Table 4.8: Novelty-oriented expansion TREC 2009 and 2010.

An informal comparison of our clusters with the TREC 2009 subtopics[10] indicates that many of the subtopics appear in our clusters, with our clusters revealing additional subtopics not appearing in the TREC subtopics. For example, Figure 4.6 lists the official subtopics provided by the TREC Web Track organizers for topics 15 and 33 corresponding to the clusters from our result shown in Tables 4.6 and 4.7 respectively. Each topic comprises a query, a general description of the query, and a number of subtopics, each indicating a different aspect or interpretation of the query. The official subtopics for topic 15 exhibit reasonable overlap with our clusters, with specific subtopics related to basketball, football, baseball, and college sports. On the other hand, in contrast to our clusters, the official subtopics in topic 33 do not focus on specific brands, but rather on different aspects of purchasing and using an elliptical trainer.

Unfortunately, a direct quantitative comparison against the method of Radlinski et al. (2010), which provided the raw material for the creation of these topics, requires the availability of commercial search engine query logs. Nonetheless, our informal comparison suggests that our approach might reasonably replace, or augment, the method of Radlinski et al.

To provide a quantitative evaluation of our approach, we used the clusters to implement a simple form of novelty-oriented query expansion. For each query, we selected the top four clusters, and created four expanded queries by adding the top three terms from each cluster to the original query. We executed these four expanded queries against the full ClueWeb09 collection indexed using the Lemur toolkit[11] with stopwords removed but no stemming.

We then merged the four results into a single result using a round-robin approach, which is known to be simple and effective (He et al., 2011). We compared the merged results against the original query using the primary TREC 2009 and 2010 Web Track novelty measures. Despite the simplicity of the query expansion and merging methods, as can be seen in Table 4.8, at depth 5, significant improvements are seen for all measures. At lower depths, improvements are still positive but do not cross the threshold of significance, except for ERR-IA@20 in 2009.

Even though these runs used only document content, and did not incorporate field weighting, link analysis, or other Web-oriented techniques, the runs would place in the top 4 at TREC 2009 (Clarke et al., 2009) and well above median at TREC 2010 (Clarke et al., 2010). The

[10] see `http://trec.nist.gov/data/web/09/wt09.topics.full.xml`

[11] `www.lemurproject.org`

performance of our expansion run would place it among the top six TREC 2009 groups for the full ClueWeb09 collection (Clarke et al., 2009), a reasonable outcome for a run that does not employ spam filtering, link analysis, or other Web-oriented techniques.

4.3.6 Summary

We have demonstrated that anchor text usage for mining subtopics is promising. Our method uncovered various subtopics for a given query using anchor text and the out-links between anchor text and their target documents. Our approach to uncovering subtopics and query intents uses a graph of *anchor text-target document* to create term clusters that help to identify various aspects and interpretations underlying ambiguous and under-specified queries, avoiding the need for search engine user interaction logs. In contrast to content-oriented methods based on clusters of top ranking documents, our approach incorporates information from across the collection through its analysis of anchor text. Moreover, our method does not depend on structured sources, such as Wikipedia, which may not reflect the diversity actually present in a collection.

Chapter 5

Soft Links: Fast, Effective and Robust Traceability Links

This chapter contains details of our work in the area of traceability links maintenance in frequently-edited documents.

5.1 Introduction

The goal of traceability link recovery and management is to recover and maintain a link between two documents such as an initial version of a document (source) and its subsequent versions (target). Maintaining a link for a specific edit between source and target may be achieved by annotating the location of an edit on the target with a unique tag every time a change occurs at the particular location. Invariably, such unique annotation tag becomes the link between the source and target documents. This type of links that are manually generated and manually maintained are referred to as *hard links*. Maintaining hard links can quickly become burdensome on the maintainers as a result of several changes, especially when portions of documents are moved from one document to another document as a result of source code refactoring or article clean-ups that are usually done in frequently edited documents such as the Wikipedia application.

Our goal is to explore the use of *soft links* as an automatic method for maintaining obsolete

and broken hard links. We provide an algorithm that automatically generates soft links for specified edit locations on the source documents. Soft links are generated from a signature of terms surrounding an edit location. A *signature* consists of terms that uniquely identify an edit location on the source document. We make use of a fast passage retrieval algorithm (Clarke et al., 2001; Clarke and Terra, 2004) as our document similarity and retrieval function where the terms in a soft link signature becomes the input query.

In software artifact maintenance, an analyst usually investigates smaller fragments of a whole document at a particular time. Passage retrieval supports a fine-grained text fragment retrieval unlike a traditional document retrieval which processes the entire contents of documents. Passage retrieval has been actively used for text segmentation, text summarization, question answering, and document retrieval purposes in information retrieval. We intend to use a *sliding window* based passage retrieval algorithm which has been previously used for question answering tasks in several TREC tracks. Details of the passage retrieval algorithm are described in Clarke et al. (2001) and Clarke and Terra (2004).

In sum, we apply document similarity to the problem of maintaining — i.e., identifying and recovering — traceable links in documents that are edited frequently. We explore a generalized framework for hypertext links recovery in frequently edited documents using the Wikipedia and source code as specific cases of such documents. Next, we describe the motivation behind the work we present in this chapter.

5.1.1 Motivation

We consider the problem of implementing fast and effective hypertext links to specific locations within documents, especially in the absence of explicit markup or complex tool support. Current methods, such as named anchors in HTML, may provide only restricted versions of these links, or may be tied to specific formats and systems. Motivated in part by the need for requirements traceability in software engineering, we wish to provide support for such links across a wide variety of document formats, including source code. For example, given source code that implements a particular software requirement, we may wish to establish a link from the location in a text document where the requirement is specified to the location in a source code file where the requirement is implemented.

When documents are static, remaining unchanged after their initial creation, a link to a specific location in the document may be effected by a byte offset, or by a similar direct reference to the target location. When documents are dynamic, these direct references must be carefully managed and mapped to reflect the changes that have been applied to the document. If the linked portion of the document is moved to another document, the byte offset or reference must be modified to follow the material. Such careful management may require that all changes flow through a closed set of tools, which are aware of these links and are designed to maintain them correctly. Any casual use of an editor, development environment, or word processor outside this closed tool set could quickly invalidate the links.

A link to a specific location may also be effected by creating and adding a unique tag at the target location, provided that the document format supports the inclusion of such tags. Locating the target of the link involves a search for the tag, which may be supported by an index over the collection of documents. If the tags are visible to users, and their significance is understood, no tool support is required to support the links, and any tool may be used to edit the documents. However, as documents are changed, and material is moved from one document to another, the users themselves must take care to maintain the relationships between the tags and the linked material, which may place a substantial burden on them.

The use of tags, offsets, and other direct references all represent forms of *hard links*, where the target of the link is unambiguous. We describe and evaluate the performance of an algorithm for generating *soft links* to specific locations in documents. We generate these soft links by creating a *signature* from the text surrounding the target location. Each signature consists of a set of words and character *n-grams* — a sequence of n characters extracted from the text. The signature is chosen to provide a balance between efficient access and its ability to robustly specify a document location.

Soft links are *resolved* to find the target location, by treating the signature as a query to determine the text that best matches it. An existing information retrieval algorithm is enlisted and extended to implement an efficient and effective version of this matching process (Clarke et al., 2006, 2001). Because the algorithm does not require an exact match between the signature and the text, the algorithm is robust in the face of edits. In essence, the algorithm treats the text surrounding the target location as a passage to be retrieved, creates a signature for that passage, and resolves the link by searching for the passage. Even if the passage undergoes editing, enough

of the signature often remains to allow the target location to be determined.

Soft links may either be used alone, or they may be combined with hard links, providing a backup when hard links fail. For example, a system might represent and store hard links in the form of filenames plus byte offsets, providing an appropriate editing or development tools to maintain them as changes are made. If uncontrolled changes are made by outside tools, which are unaware of the hard links, the soft links may help to recover the target locations and restore the hard links. However, since our focus in this chapter is the implementation and evaluation of soft links, in the remainder of the chapter we assume that soft links are the sole method available.

After a review of related work in Section 5.2, we summarize the existing algorithm and describe the extensions required to implement soft links in Section 5.3. In Sections 5.4 to 5.7 we present a series of experiments to establish the efficiency and effectiveness of our soft links algorithm. These experiments are based on three collections: two large source code collections (Apache Ant Core and Linux kernel) and one large text collection (Wikipedia). The experiments examine our ability to take soft links established in one version of the collection and use them to find the same target locations in later versions of the collections, after years of changes. The paper concludes with a summary and discussion of future work (Section 5.8).

5.2 Background and Related Work

5.2.1 Background

The area of software requirements traceability provides a major inspiration for our work. Most efforts in software traceability have focused on the discovery of traceable links for requirements documents, source code, and other software artifacts. As software systems evolve, care must be taken that traceable links are not rendered obsolete on subsequent versions of the same artifact. This requirement introduces an additional level of challenge into the software maintenance process. Maintaining traceable links for evolving software is one of the fundamental issues in traceability link management (Gotel and Finkelstein, 1994; Jiang et al., 2008), often requiring substantial database (Mundie and Hallsworth, 1995) and automated tool support (Ramesh, 1998).

The Requirements Traceability Matrix (RTM) represents one important tool for relating software artifacts. When a change request occurs, software engineers may be required to manually update an RTM in order to document the change and establish a trace link for the affected artifacts. Problems associated with the manual update process include link degradation and excessive complexity (Cleland-Huang et al., 2005). Manual maintenance effort for an RTM becomes prohibitive for large-scale projects due to the sheer number of features and dependencies existing in these projects. As a result, it is not uncommon for traceability matrices to quickly become obsolete (Hayes et al., 2003).

5.2.2 Related Work

The problem of maintaining traceable links in frequently edited text such as source code and Wikipedia has been well recognized and different researchers have proposed different solutions. Murta et al. (2006) and Mader et al. (2008) use policy-based approaches for traceability link maintenance. When a change event occurs in the system, a policy is automatically selected to perform the actions necessary to evolve the traceable links. Jiang et al. (2008) compare versions of source code to recover and update traceability links using the latent semantic indexing algorithm. Hammad et al. (2009) also compare versions of source code in order to update traceability links in software artifacts and class diagrams. In contrast to our algorithm, these approaches are designed for specific software engineering contexts having careful version control and change tracking functionality. They may not necessarily translate across a wide range of environments and document formats.

Origin analysis (Godfrey and Zou, 2005) is a method for inferring the basis for design changes in source code. It provides insights into why elements in one version of code have changed in other versions of the same code. It is a process used for inferring the non-documented contexts and reasons for design changes between source code versions. When traceable links point to specific code elements, the result of origin analysis can be used for inferring reasons behind code evolution. Godfrey and Zou (2005) utilized origin analysis to detect splits and merges in source code. They proposed a semi-automatic process to obtain precise results with the help of human developers. In contrast, our method does not attempt to provide reasons and contexts for code evolution.

Kim et al. (2005) utilized the similarity between source code functions to perform automatic function tracing between versions of source code artifacts written in C programming language. Their function similarity method combines the function name, function calls, keywords and variable names in the code, and a composite complexity metric. Therefore, their method can still locate function names that have been changed between source code versions by using their similarity measure. They reported 87.8% accuracy on the Apache 2 Web Server project in their consideration. In contrast with our method, their method is specifically designed for source code artifacts which are written in C programming language. Our method also supports the tracing of arbitrary locations, unlike their method which performs only function-by-function tracing.

Hayes et al. (2003) frame the requirements traceability task as an information retrieval problem, with the goal of improving the recall and precision values for candidate traceability links. They implemented several retrieval algorithms on the MODIS data set, provided by NASA, comparing the performance of their methods against manual judgments. The RETRO traceability tool (Hayes et al., 2007) extends this approach to support the automatic generation of a requirements traceability matrix. Generation of the requirements traceability matrix is outside the scope of our research problem. Antoniol et al. (2002) investigated probabilistic and vector space approaches for linking C++ source code to manual pages and Java source code to functional requirements. In this thesis, we implement traceability links using a *passage-oriented* information retrieval algorithm, allowing us to easily specify links to target locations within source code and other documents.

Duala-Ekoko and Robillard (2007) employed a related idea to track evolving code clones. They obtain their code clone input from the output of SimScan[1] which performs the actual clone detection by scanning the entire source code repository. Thereafter, they track the evolution of detected clones by combining the similarity between code clones, as well as their structural and lexical layout information to represent individual code clone regions. Therefore, when filenames and byte regions of clones are changed, their representation of clone regions are resilient to changes and can still be utilized to locate the clones.

In contrast to our work, their primary goal is to track the evolution of code clones, our method and implementation has a broader goal that locates any edit location in evolving artifact, whether

[1] `http://blue-edge.bg/simscan`

they are clones or not. Their implementation is specifically built for source code written in Java such that specific Java constructs are hard-coded into their code clone tracking method. Our method may be applied across source code written in various programming languages. Since their method tracks pre-determined clones identified by SimScan, newly introduced clones will not be included in their tracking until another clone detection scanning operation is carried out by SimScan. Unfortunately, by scanning the entire repository over again, the efficiency of the system degrades. Moreso, they reported it took 32 minutes to scan the 63 kLOC in their experiments. Our algorithm utilized inverted index to achieve very fast processing in all our extensive traceability experiments.

Our traceability method is applicable to adhoc traceability tasks in which the edit location is not pre-determined but a software maintenance engineer may select the edit location on the fly. As reported in our result, our method performs a trace on a significantly larger artifact within an interactive time limit. Apart from source code, our method can also perform trace on other forms of software artifacts such as requirements documents. Since they did not report any of the standard performance metrics such as precision and recall, it is difficult to do a direct comparison of the performance between their method and ours.

Related to our work, but outside the area of requirements traceability, Golovchinsky (1997) and Golovchinsky and Chignell (1993) present a browsing-as-a-search paradigm. Their work explores a bridge between document search, where users provide search terms, and browsing, where no explicit search terms are provided. Users often have problems in selecting the best terms to represent their search intentions. However, these intentions may sometimes be described with either a paragraph in a document or the entire document text. This work explores the automatic selection of discriminating query terms from a user's browsing session, with the search engine using selected query terms as links for retrieving similar documents.

Other work explores the connection between queries and hypertext links, generally in the context of the Web. For example, Wang et al. (2009) focus on exploratory search, attempting to augment keyword search with browsing, where browsing is based on topic maps generated from search logs. These topic maps are related to the signatures we generate in our work. From these topic maps, Wang et al. attempt to learn the search and browsing behavior of previous users, in order to improve the search and browsing experience of future users. Olston and Chi (2003) propose an interface incorporating both keyword search and browsing. Keywords are

used to score the content of individual URL hyperlinks. Based on the relevance of the linked pages, URLs in browsed documents are highlighted and annotated to reflect the most relevant hyperlink the user should follow. Shen et al. (2006) map browsing problems into searching problems by reformulating queries, streamlining interactive search and browsing by way of a single visualization interface.

5.3 Generating Soft Links

We base our implementation of soft links on an established passage retrieval algorithm, which was originally developed as an initial retrieval step in a question answering system by Clarke et al. (2006, 2001). Given a query, consisting of words and other search terms, the algorithm searches an indexed collection of documents to identify appropriately sized passages of text related to the query terms. For example, suppose we wish to answer the question "Who used to make cars with rotary engines?" After analyzing the question, a basic question answering system might determine that the query "car rotary engine" is likely to retrieve passages containing the answer from an available collection of documents. After executing the query against the collection, the top k passages would be subjected to further analysis to extract the answer (i.e., "Mazda Motor Corp"). Depending on the complexity of the question answering system, this analysis may be substantial, possibly involving additional retrieval steps. An introduction to basic and advanced question answering systems is given by Prager (2006). Examples of advanced question answering systems include the Watson system[2], which includes passage retrieval as one of its many components. In the remainder of the chapter, we focus on passage retrieval as a basis for implementing soft links, ignoring other aspects of question answering.

An important feature of the passage retrieval algorithm developed by Clarke et al. is its ability to efficiently and directly retrieve arbitrary passages from the collection, i.e., any sequence of indexed tokens in the collection, regardless of length, sentence boundaries, etc. The algorithm efficiently returns those passages that maximize the passage scoring function appearing as Equation 5.1 in Section 5.3.1, returning the top k passages for a pre-specified value of k. Often, passage retrieval algorithms are limited to retrieving passages of a fixed size (e.g., 50 words,

[2] `www.watson.ibm.com`

three sentences, etc.). Other algorithms do not directly index tokens within a document, but rather follow a two-step process, where they first retrieve related documents and then identify passages through a post-retrieval scanning step (Tellex et al., 2003). We adapt this algorithm to implement soft links by replacing the query with a signature extracted from the text surrounding the target location.

In the remainder of this section we provide a summary of the passage retrieval algorithm (Section 5.3.1) followed by a detailed explanation of its extension to soft links (Section 5.3.2). For the experiments reported in later sections, we use the implementation of the algorithm available as part of the Wumpus open-source search engine[3]. Additional information regarding the efficient implementation of the algorithm may be found in Clarke and Terra (2004).

5.3.1 Passage Retrieval

The passage retrieval algorithm assumes the existence of a document collection $\mathcal{C}$, which may consist of text, source code, and similar data. For the purpose of passage retrieval, the algorithm treats this collection as a single long string, as if all the documents were concatenated together. For now we ignore boundaries between documents, but we will re-introduce them shortly. The algorithm further assumes that the collection has been appropriately tokenized — i.e., split into words, or other units, consistent with the language of the documents. As a result, we can view the collection as a sequence of tokens

$$\mathcal{C} = t_1 t_2 t_3 ... t_N,$$

where N is the length of the collection and t_i is the token at position i in the collection.

For example, consider the source code in Figure 5.1, which is taken from a version of the Linux kernel dated February 20, 2007. The figure shows the start of a function from the source file "`sound/sound_firmware.c`". A simple (but typical) method of tokenizing this code defines tokens as sequences of alphanumeric characters separated by sequences of non-alphanumeric characters[4]. When we tokenize this version of the Linux kernel, the first token of

[3] `wumpus-search.org`

[4] Naturally, we could be more sophisticated in our approach to tokenization. In practice, this simple method is

```
static ssize_t show_ordinals(
  struct device *d, struct device_attribute *attr, char *buf
)
{
    struct ipw2100_priv *priv = dev_get_drvdata(d);
    u32 val = 0;
    int len = 0;
    u32 val_len;
    static int loop = 0;

    if (priv->status & STATUS_RF_KILL_MASK)
        return 0;

    if (loop >= sizeof(ord_data) / sizeof(*ord_data))
        loop = 0;
```

Figure 5.1: Example code from the Linux kernel as of February 20, 2007.

Figure 5.1 appears at position 50350223 in the token sequence. A tokenization for the first few lines of the figure appears in Figure 5.2.

For passage retrieval, we represent a query Q by a set of tokens, which are matched against the tokens in the collection. For example, we might execute the query

$$Q = \{\text{“}\texttt{priv}\text{”}, \text{“}\texttt{drvdata}\text{”}, \text{“}\texttt{attr}\text{”}\}$$

against the collection illustrated by Figure 5.2. The result of executing such a query is expressed as a ranked list of *extents*, where each extent is a pair of positions in the collection $[u, v]$ indicating the start and end of a passage. An extent containing a token $t \in Q$ is said to *match* t, for which we write $t \in [u, v]$. For the example query above, the extent $[50350234, 50350243]$ matches all three tokens in the query, the extent $[50350227, 50350239]$ matches two of the tokens, and the extent $[50350224, 50350231]$ matches none of them.

The passage retrieval algorithm ranks extents according to their length and the number of tokens they match, where a token appearing multiple times in an extent — such as the token “`priv`” in the extent $[50350234, 50350243]$ — counts only once. A shorter extent is favored over a longer one, and an extent containing more matches is favored over one containing fewer. Generally, extent size is minimized as much as possible. The algorithm balances the length against the number of matches by way of the following passage scoring function (Clarke et al., 2001):

$$P(Q|u, v) = \sum_{t \in Q \wedge t \in [u,v]} \log(N/N_t) - |Q| \cdot \log(v - u + 1)\,. \tag{5.1}$$

where $P(Q|u, v)$ is the passage scoring function for Q having start position u and end position v, and N_t is the frequency of term t in the corpus.

The passage retrieval algorithm employs an index skipping technique to efficiently return the top k extents maximizing Equation 5.1 across the collection. As extents are generated, the algorithm discards those extents that cross document boundaries, since these do not represent genuine passages. In addition, the top k extents are pre-filtered to exclude all but the best scoring extent from each file. Our experience shows that without this pre-filtering the top k extents may be

sufficient as a basis for implementing soft links over English text, and can be extended to source code with character n-grams, as discussed in Section 5.3.2.

50350223	50350224	50350225	50350226	50350227
static	ssize	t	show	ordinals
50350228	50350229	50350230	50350231	50350232
struct	device	d	struct	device
50350233	50350234	50350235	50350236	50350237
attribute	attr	char	buf	struct
50350238	50350239	50350240	50350241	50350242
ipw2100	priv	priv	dev	get
50350243	50350244			
drvdata	d	...		

Figure 5.2: A simple word-oriented tokenization of the code in Figure 5.1.

dominated by a large number of extents from a single document. While in theory, the maximum size of an extent is the size of the longest document in the corpus, Equation 5.1 strongly favors shorter extents when identifying top ranking passages.

5.3.2 Signatures and Soft Links

We now extend the passage retrieval algorithm to the implementation of soft links. Unlike question answering, where a query can be derived from terms in the question, we must base the query for a soft link on terms appearing at or near the target location. This query becomes the signature for the soft link, which is resolved by the passage retrieval algorithm to identify the target location. We aim to generate a signature that identifies that location and no other. Although the passage retrieval algorithm returns an extent, rather than a single position in the collection, the nature of the passage retrieval algorithm encourages a short extent, with the target location contained within it.

During our preliminary experiments with soft links, we realized that our simple word-oriented tokenization might not be sufficient to meet the needs of soft links into source code. For example, identifiers in source code are often long relative to words in English text. Moreover, while identifiers are often chosen for their mnemonic value, their morphology is unconstrained by the rules of any natural language. The token "`drvdata`", appearing in Figure 5.1, could be renamed to "`driverdata`" throughout the source code without violating any programming language rule or compromising its mnemonic value. As a result of this experience, we implemented soft link signatures using a combination of word-oriented and character n-gram tokens. The character n-gram implementation performs as effective as word-oriented soft link. Hence, we decided to implement soft link signatures with character n-gram tokens because it is amenable to languages having unknown or complex morphology such as source code.

Character n-grams are constructed by splitting a string into overlapping tokens of n characters each. For example, we might split "`drvdata`" into the character 4-grams: "`drvd`", "`rvda`", "`vdat`", and "`data`". These n-gram tokens could then be indexed at the position of the original word-oriented token. In the case of our example, they would be indexed at position 50350243. For the experiments reported in later sections, we index both word-oriented tokens and 4-gram

1. Given the target's position i in the collection, construct a window of size w tokens around the target: $[i - \lfloor w/2 \rfloor, i + \lceil w/2 \rceil]$.
2. Form a set W of terms from the word and n-gram tokens appearing in this window.
3. Rank all $t \in W$ according to N_t/N in increasing order. The top m terms become the candidate signature Q
4. Resolve Q against the collection.
   (a) If the candidate signature uniquely returns the target as the top scoring result, the candidate signature becomes the soft link.
   (b) If the candidate signature returns multiple extents with the same top score, including the target, check for duplicate content.
   (c) Otherwise, increase either m, w, or both m and w and retry steps 1-3.

Figure 5.3: Summary of soft link generation procedure.

tokens derived from them. Note that we chose not to allow n-grams to cross word boundaries, although in principle these tokens could also be included in the index and used in signatures.

Character n-grams are known to be an effective method for information retrieval tasks over collections with unknown or complex morphology (McNamee and Mayfield, 2004; McNamee et al., 2008). In our experiments, we set n to 4 based on results obtained by McNamee and Mayfield (2004), who demonstrate that this is a reasonable value for many languages. In the case of source code, the use of n-gram tokens in signatures can allow partial matches when identifiers change. For example, the 4-gram "`data`" in a signature would still provide a partial match if "`drvdata`" were renamed to "`driverdata`".

We follow a simple procedure to generate signatures. Given a target location at position i in a given source document, we start with a window of text of size w centered on the token t_i. Tokens within the window need to be in close proximity to the target location. For the experiments reported in later sections we use $w = 100$, a value that produced reasonable results in our preliminary experiments. We then form a set W of terms from the tokens appearing in this window, and select from it a subset of size m, which becomes a *candidate* for the signature. To select this subset, we rank the elements of W according to their relative frequency in the collection (N_t/N), under the assumption that rarer terms will do a better job of uniquely locating the target location. For the experiments reported in later sections we use values for m between 1 and 20. In our preliminary experiments, values for m greater than 20 provided little or no improvement in any of our effectiveness measures (an observation which is confirmed by the experiments reported in later sections).

Once we have a candidate signature, we test it against the collection. If the candidate signature uniquely returns the target as its top result, with all other extents having lower scores, the candidate signature becomes the signature for the soft link. If the candidate signature fails to uniquely locate the target, we can increase either m, w, or both m and w and retry, generating a new candidate. In some cases, the collection may contain identical documents or documents with long chunks of identical content. In these cases, we may have to provide special handling or simply accept the fact that a soft link may have multiple valid targets. The treatment of duplicate content depends on details of the application environment. Figure 5.3 summarizes the soft link generation procedure.

Type	Collection	Version	Size	Indexing time
Text	Wikipedia	30-Nov-2006	18.9GB	26.36 minutes
		03-Nov-2009	33.18GB	56.42 minutes
Code	Apache	03-Feb-2007	39.85MB	424 ms
		25-Mar-2010	43MB	427 ms
	Linux	20-Feb-2007	456.7MB	6.527 seconds
		25-Feb-2010	718.8MB	9.418 seconds

Table 5.1: Indexing time for data sets.

Type	Collection	Version	Size	Size including n-grams
Text	Wikipedia	30-Nov-2006	5.86GB	10.5GB
		03-Nov-2009	18.9GB	33.18GB
Code	Apache	03-Feb-2007	16.3MB	39.85MB
		25-Mar-2010	17.6MB	43MB
	Linux	20-Feb-2007	207.8MB	456.7MB
		25-Feb-2010	324.1MB	718.8MB

Table 5.2: Collections used in our basic experiments.

Dataset	% Documents Changed	% Hard Link Success
Wikipedia	97.2	8.1
Linux Kernel	93.4	20.9
Apache Ant	36.0	76.4

Table 5.3: Evolution of the collections used in our basic experiments, based on 1000 randomly selected documents and randomly selected locations in each document.

5.4 Basic Experiments

In this section, we provide details of experiments illustrating basic properties of the algorithm, including basic measures of efficiency and effectiveness.

5.4.1 Data Sets

The choice of our collections reflects the fact that our algorithm must be suitable for both source code and text, Table 5.2 provides information about each of the test collections used in our experiments. They are all widely known and generally available open source and copyleft projects. For each collection, there is an initial version and an evolved version. The initial version represents the state of the collection at a given point in time, and the evolved version represents the state of the collection after several years of frequent and on-going changes.

One project, the Wikipedia, consists primarily of text. The other two projects, the Linux kernel and the Apache Ant Core, consist primarily of source code, with one written in the C programming language and the other written in Java. For these two projects, we removed non-source files, retaining only the `.java` files from the Apache Ant Core, and the `.c` and `.h` files from the Linux kernel. The comments in the source codes are preserved. We consider two different programming languages in order to examine the programming-language independence of our solution. Each of these collections is far larger than many similar collections of source code and text, helping us to test the efficiency and effectiveness of our algorithm under relatively harsh circumstances. Table 5.1 shows the time it takes to build the inverted indexes using the Wumpus system. We see from the table that it takes less than ten seconds to index each source code data set and less than one hour to index the largest Wikipedia collection of 33.18GB[5].

Table 5.3 illustrates the evolution of the collections used in our basic experiments. To measure evolution, we randomly selected 1000 documents from the initial version of each collection. For the Wikipedia collection, 97.2% of the selected documents exhibited changes between the initial and evolved versions, based on an exact byte-for-byte comparison. For the Linux kernel, 93.4% of the selected documents changed; for the Apache collection, 36.0% of the selected documents

[5] see Tables 5.1 and 5.2

changed. As a second test, we randomly selected byte offsets within each of the files selected from the initial collection. As discussed in the introduction, the combination of a file name and a byte offset represents a simple method for constructing hard links. For each of these hard links, we assessed them to determine which still link to approximately the same source code or text in the evolved collection (allowing for minor shifts in the text.) Out of the selected documents and edit locations, only 8.1% and 20.9% correctly linked to locations in Wikipedia and Linux Kernel projects respectively, while the Apache Ant project had a much higher success rate of 76.4%. The hard link success values shown in Table 5.3 may be compared to the precision@1 values shown in later experiments.

5.4.2 Experimental Methodology

Our basic experimental methodology selects a random position in a random document from the initial collection as the target (i.e., edit location) for a soft link. Tokens in the neighborhood of the selected targets are then used to construct a soft link according to the algorithm described in Section 5.3. We then attempt to resolve the soft link against the evolved collection, determining if we can successfully identify the same target after a series of modifications. We repeat this process for $x = 1000$ random positions from each collection, evaluating the overall success using the measures defined in the next subsection.

Success is determined in several ways, both through relatively crude automatic judgments and through more careful manual judgments. First, recognizing that the targets of many of the soft links will in fact have remained in the same document (e.g., Wikipedia page name or source file path) throughout the evolution of the collection, we consider it a success if we correctly identify the file in the evolved collection that contained the soft link in the original collection. While this is a simplistic definition of success, it allows us to easily obtain a picture of the algorithm's performance over thousands of soft links. For the two source code collections, we also employ a slightly less simplistic definition by tracking the method or function name containing, or closest to, the target of the soft link. When reporting experiments, we refer to these definitions as *path* and *block* definitions, respectively. Note that the block definition is more restrictive than the path definition, since a soft link that succeeds according to the block definition will always succeed according to the path definition, except if the function or method has been moved to another file.

Naturally, these automatically judged definitions miss one of the most useful aspects of soft links, namely the ability to identify targets that have moved from one file or document to another. To determine success in these cases, we manually judge soft links that did not succeed according to the path and block definitions. For each such soft link, we assess whether its target has moved, the target has been renamed, the target has been deleted, or the soft link algorithm was unable to recognize the target of the soft link. Only in the last case has the algorithm truly failed.

In these experiments we work with a fixed neighborhood having a window size of $w = 100$ tokens. For a very small number of the randomly selected targets (much less than 1%) a soft link generated from this neighborhood does not uniquely define the target, often due to duplicate content appearing elsewhere in the collection. As we discussed in Section 5.3, in practice the window size would be increased or the duplicate content would be flagged, depending on details of the application environment. In order to avoid this complexity, for the purposes of these experiments we discard these targets and randomly select replacement targets. We leave the implementation of dynamic window size which would include additional highly discriminating tokens in the signature as future work.

In these experiments, we use both words and character 4-grams as tokens. For each target, we generate a signature from the top m terms appearing in the window, according to the ranking function given in Section 5.3. We explore the performance of our algorithm for values of m between 1 and 20.

5.4.3 Performance Measures

We consider both the efficiency — the time it takes to resolve soft links — and effectiveness — our ability to resolve soft links against the evolved collection. To measure efficiency, we record the elapsed time from when a query is issued to when it completes on a standard desktop machine, circa 2010. To measure effectiveness, we borrow and adapt a number of standard measures from the field of information retrieval (Büttcher et al., 2010).

Our most important measure is *precision at rank 1* (precision@1), which reflects the algorithm's ability to return the target of the soft link as the top ranked result. If the algorithm returns the target of a soft link as the top ranked result, it receives a score of 1; otherwise, it receives a

score of 0. The value of precision@1 is the average of these scores across a set x of soft links, where $x = 1000$ for these experiments.

If the target of a soft link is not returned as the top result, a software engineer recognizing this problem might still be able to recover the target by examining the rest of the top k extents returned by the algorithm. This aspect of the algorithm's performance can be measured by *success at rank* k (success@k).

Since there is only one correct target for a given soft link, we assign a score of 1 if the algorithm returns the target in the top k; otherwise it receives a score of 0. The value of success@k is the average of these scores across a set of x soft links. For these experiments we report success@20. Since there is only one correct target for a soft link, we note that success@1 is the same as precision@1. We use distinct names for clarity and to emphasize the different roles these measures play. Since success@20 includes cases where the target is returned as the top result, we define an additional measure, which we call *recovery at rank* k (recovery@k). This measure is defined as success@k for those soft links where the target is not returned as the top result. These are the cases for which the intervention of a software engineer would be required to recover the target. For these experiments we report recovery@20.

In addition, we report *Mean Reciprocal Rank* (MRR), a standard effectiveness measure for ranked retrieval in information retrieval when only a single result in a ranked list can be correct. MRR is simply the reciprocal of the rank at which the correct result appears averaged over the set of $x = 1000$ soft links. A perfect MRR value is 1.0.

5.5 Results and Discussion

In this section we report results of our basic experiments over the collections listed in Table 5.2.

5.5.1 Effectiveness measures

Figure 5.4 presents the success@20 values for soft link signatures containing various numbers of query terms ($1 \leq m \leq 20$) for both path and block definitions. Figures 5.5 and 5.6 present

the corresponding precision@1 and MRR values, respectively. Figure 5.7 presents recovery@20 values, indicating the performance of the algorithm when the top link is not correct.

Overall, values for Apache are superior to those obtained for both the Linux kernel and Wikipedia projects. We attribute this difference to two phenomena: (i) the smaller size of the Apache source code has a positive impact on effectiveness, and (ii) there are fewer changes between the original version of Apache and its evolved version. As expected, results for the stricter block definitions fall below those for path definitions in all measures except recovery@k. The attention of a software maintenance engineer is required to recover a target mostly when methods and functions have moved across files. Therefore, more links are correctly located for this category of edit, i.e. block level edits. In the next subsection, we manually investigate reasons for these failures, which may result from legitimate changes, such as movements and deletions, as well as from outright failures of the algorithm.

5.5.2 Failure analysis

In order to understand why some of the soft links fail under the cruder automatic assessments, we manually assessed 200 failed results for each of the Linux kernel and Wikipedia collections, including all failures at the $m = 20$ level (i.e., with 20 query terms.) This manual assessment provides us with an indication of reasons for failure. To perform this manual assessment, we obtained files in both earlier and later versions of the corpus. Contents of the files are reviewed to determine whether there has been code or article deletion, code or text movement between files, or file or article renaming.

Figure 5.8 summarizes the result of our manual assessment. In general, the algorithm successfully detected movement, deletions and similar changes, accounting for most failures under automatic assessment. Only in the case of Wikipedia did the algorithm itself fail. About 60% of the failures in the Linux kernel were due to code deletion. Another 30% of the failures were due to code being moved from one file to another. The remaining 10% of failures were due to file renaming, especially from one subdirectory to another subdirectory.

On the Wikipedia collection, we observed that about 80% of the failed results were due to renamed article titles. For example, the article titled “Chippewa mythology” in the 2006

Wikipedia was renamed to "Anishinaabe traditional beliefs" in the 2009 Wikipedia. Similarly, "Windows Sound Recorder" was renamed to "Sound Recorder (Windows)", and "Mesa College Foundation" was renamed to "San Diego Mesa College". Roughly 7% of the failures were caused by the movement of text from one article to another. Another 7% were caused by article deletion. The remaining 7% were actual algorithm failures, where there had been enough edits near the target to break the soft link.

The reader should keep in mind the results of this manual assessment when considering the results from the automatic assessments. We view these automatically assessed results as providing a rough lower bound on performance — suitable for tuning parameters, determining asymptotes, and gaining an overall sense of performance — which can be easily replicated on any collection. Based on these results, we recommend a value of $m \geq 15$, which should be sufficient even for collections as large as Wikipedia.

5.5.3 Analysis of n-grams

As an additional step in our analysis, we tested the value of including n-gram tokens in the source code index. The use of n-gram tokens was suggested by our preliminary experience and by standard information retrieval practice. We repeated the basic experiments reported in this section over Linux kernel and Apache Ant collections using only the simple alphanumeric tokens described in Section 5.3.1, i.e., without n-gram tokens. Somewhat to our surprise, the overall results were essentially unchanged across all effectiveness measures. We suspect that this result reflects the relative lack of changes to variable, function and other naming conventions as the collections evolved. Nonetheless, we continue to recommend the use of n-grams for source code, which do no harm in these experiments, and may provide some additional robustness in other situations.

5.5.4 Efficiency measures

To resolve a soft link, we run the signature against the indexed collection. The Wumpus IR system is used to build the inverted indices. As reported earlier in the chapter, it took less than 10 seconds to build inverted indices for source code corpora, and less than one hour to build

the largest Wikipedia index. Using Wumpus, it is possible to incrementally update the inverted index in order to accommodate document additions and deletions without re-indexing the entire corpus.

We expect that the average time taken to resolve a soft link will increase as a collection evolves over time. As signature terms are dropped during editing, the algorithm may be forced to work harder in order to locate the top scoring passages. Figure 5.9 presents average retrieval times over the original collections; Figure 5.10 presents average retrieval times over the evolved collections. Note the difference in the scale of the y axis. For the source code collections, average resolution time is less than 10ms on the original collections and less than 50ms on the evolved collections. Even on the much larger Wikipedia collection, average resolution time is less than 70ms on the original collection and less than 300ms on the evolved collection. These times are all within reasonable bounds for interactive use.

5.6 Example Application

A possible criticism of the basic experiments is that the target positions are selected randomly, which may not coincide with the locations where the targets of soft links might actually be required in practice. In this section, we present experiments and results for an example application that illustrates the effectiveness of our soft link algorithm in a more realistic context. In this example, we trace the locations of specific source code changes from an original version of a software system to their locations in an evolved version of the system. Our idea is that these locations might reasonably be the target of soft links arising from bug reports and change requests that indicate reasons for the changes. We identify the changes by applying the *diff* utility to successive releases of the system, i.e., between the original system and its next release, generating soft links to each of these changes. For simplicity, we focus on changes in which code was added or replaced. We then resolve these soft links against an evolved version of the system, after several years of additional changes.

The Linux kernel was used for the experiments. Committed changes to the kernel were obtained using *diff* between versions `linux-2.6.20.1` and `linux-2.6.21.1`, dated February 20, 2007 and April 27, 2007 respectively. Figure 5.11 shows an example of one such change

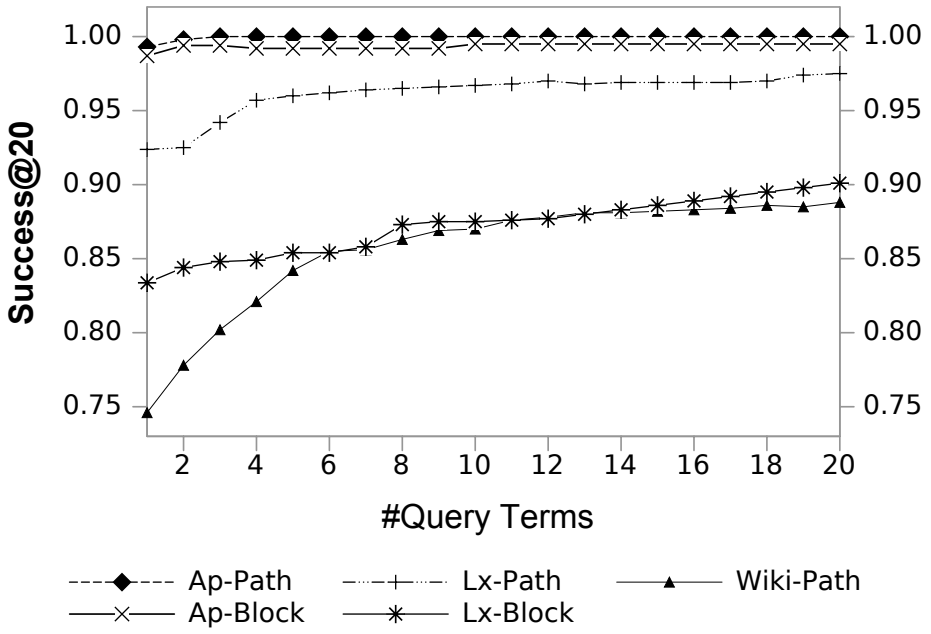


Figure 5.4: success@20: All Collections

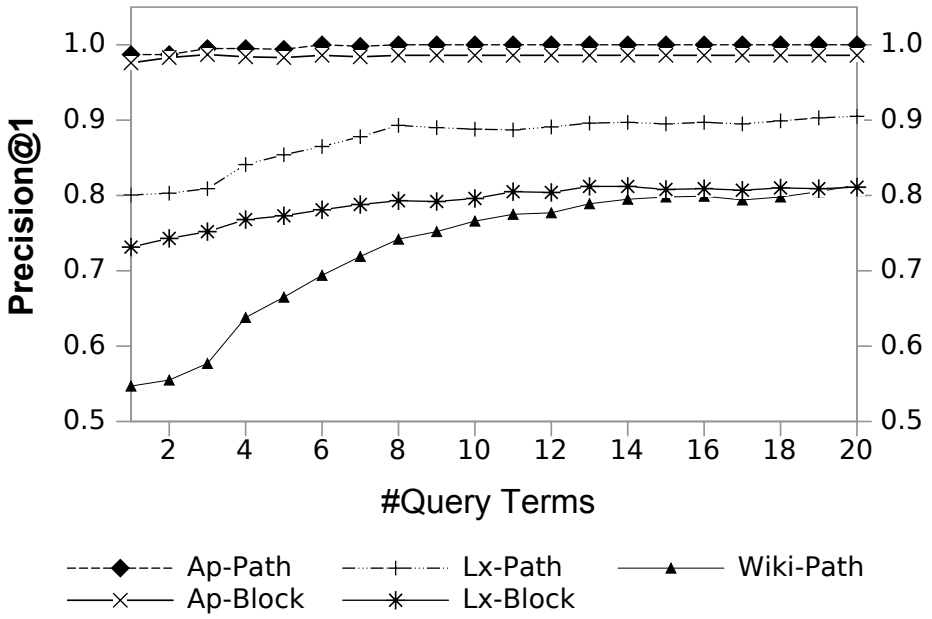


Figure 5.5: precision@1:All Collections

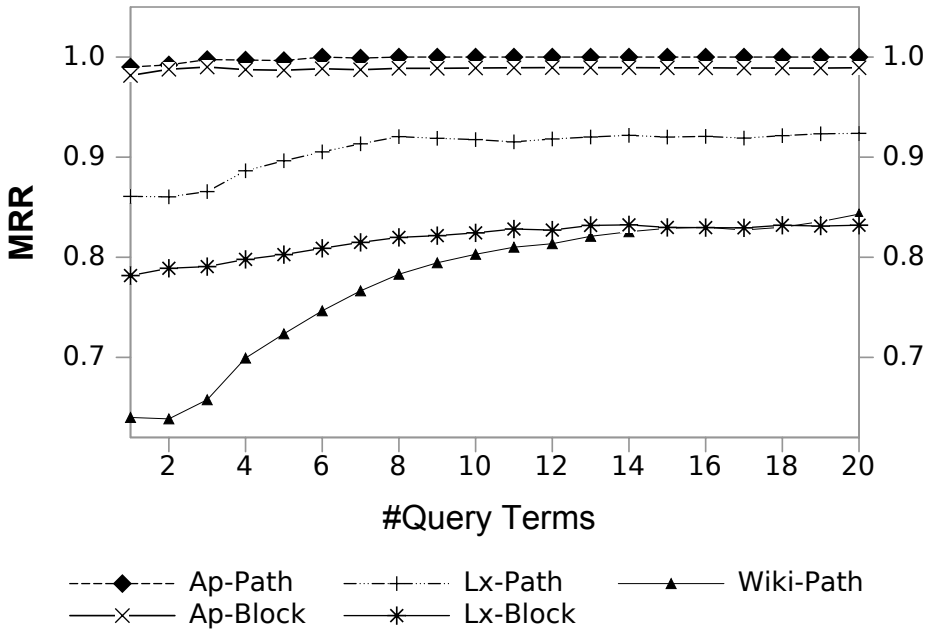


Figure 5.6: MRR: All Collections

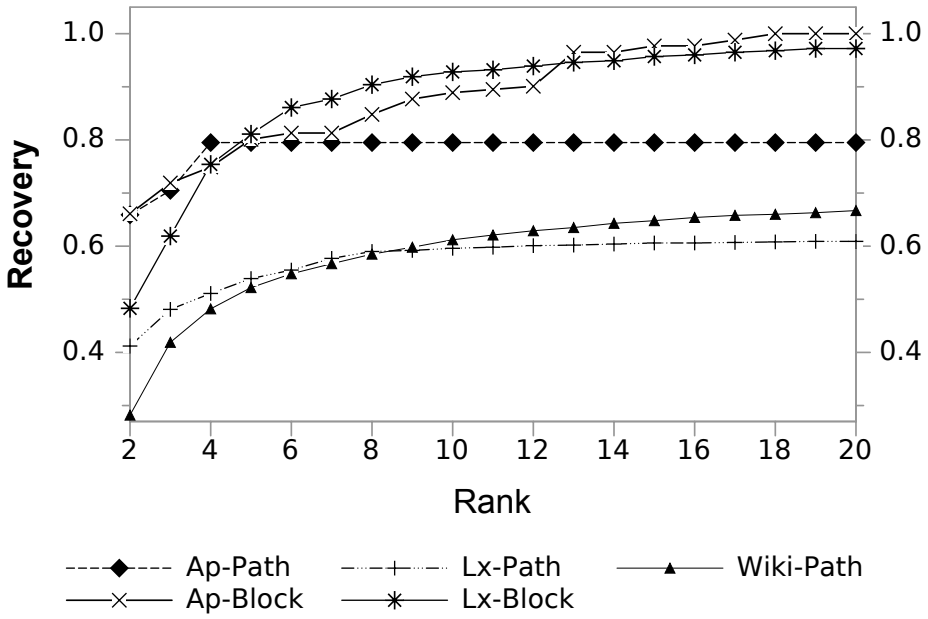


Figure 5.7: recovery@20: All Collections

in which the last two lines in Figure 5.1 was changed. In total, there were 452 locations where code was added or replaced. Soft links to these target locations were generated and resolved against version `linux-2.6.33` dated February 24, 2010. As we did in previous sections, we evaluate the results using a combination of automatic judgments and manual failure analysis.

Of the 452 targets, 349 (77.2%) of the soft links resolve to the same file in the evolved version as in the original version (i.e., precision@1 = 0.772 which is close to 0.811 obtained in our previous basic experiment.) Of the 103 soft links that do not resolve to the same file, 47 of them rank the file between ranks 2 to 10. A manual failure analysis of the remaining 56 soft links reveal that only 2 are actual failures of the algorithm, where the code remains in the system but the algorithm fails to locate it in the top 10 ranks. For 18 (32%) of the soft links, the code was deleted from the system. For 36 (64%) of the soft links, the file name or path had changed as the system evolved.

5.7 Robustness

As a final experiment, we examine the robustness of soft links as terms are deleted from them. By *robust* we mean the degree to which a signature can survive the deletion of terms, continuing to correctly resolve the soft link to the target location. Understanding how the deletion of terms impacts a soft link provides insight into how edits that remove these terms would impact the soft link. For this experiment, we return to the collections and methodology of Section 5.4.

Basically, we ask how the deletion of j of m terms from the signature will impact the soft link's ability to resolve the target location, for $0 < j < m$. We start with the 20-term signatures ($m = 20$) used in Section 5.4, which contain both word and character 4-gram terms. These signatures are constructed from the top $m = 20$ terms in a window of $w = 100$ tokens surrounding each target location. To test the robustness of these signatures, we successively delete terms from them, starting with the best scoring and top ranked term. As each term is deleted, we measure precision@1. The results are presented in Figure 5.12.

These results may be compared to those in Figure 5.5. That figure shows how precision@1 increases as top ranking terms are added to the signature, i.e. as m is increased from 1 to 20. In contrast, Figure 5.12 shows how precision@1 decreases as the top ranking j terms are removed

from a signature of size $m = 20$, for $0 < j < m$. The figure provides a rough indication of the algorithm's behavior if edits to the text had deleted these terms. As terms are removed, precision drops. However, the drop does not become sharp until 14 or more terms are removed. Apache in particular still obtained a high precision@1 value even for a single term. Once again, we attribute this phenomenon to the relatively lower rate of evolution that occurred in it. The smaller size of the collection is another factor.

5.8 Summary

We have presented an implementation of soft links for supporting links maintenance in frequently edited documents. Our method is fast and effective and it is suitable for large collections of both text and source code. Our implementation builds on an existing passage retrieval algorithm, which treats the text surrounding the target of a soft link as a passage to be retrieved, creates a signature for that passage, and resolves the link by searching for the passage. We carried out an extensive evaluation for both the efficiency and effectiveness of our soft links method on three very large projects.

Under ideal circumstances, if hard links could be carefully maintained across edits, soft links would not be necessary. For example, if hard links are implemented as file names and byte offsets change, a software development environment or text editor could maintain a link by adding and subtracting from the offset as preceding text is inserted and removed, changing the file name if the text is moved to another file, and discarding the link if the text is deleted. If hard links are implemented with unique tags, a developer or author seeing a tag could carefully retain its association with the appropriate text as changes were made, provided that the developer or author was aware of the tag's significance. In practice, such approaches are fragile, easily undermined by unaware people and tools.

Our original inspiration for this research came from problems in requirements traceability, and we intend to more thoroughly explore this application area in future research. Apart from the experiments reported in Section 5.6 our evaluation methodology is somewhat artificial, depending on randomly selected target locations, and we plan to extend our work to more realistic data sets and environments. The current work validates the core idea: that soft links can be

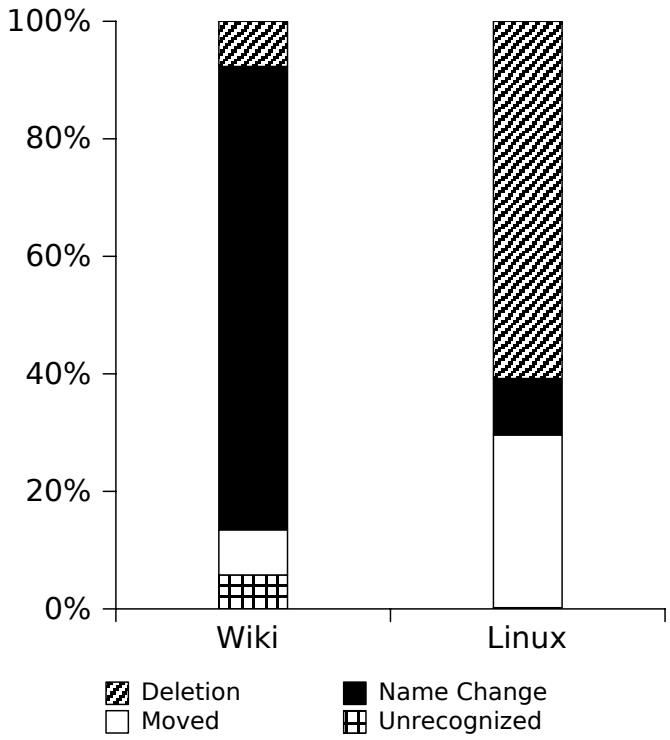


Figure 5.8: Failure Analysis: Linux kernel and Wikipedia

used as a robust substitute for hard links. Substantial additional work (in multiple environments) would be needed to truly understand all the problems and issues that arise in practice. One next step for this work is to try it out in a realistic case study to determine if the method is of actual value to software engineers for traceability and other purposes. Another potential use is to complement soft links with hard links, warning the software engineer when a hard link has failed, so that the software engineer can use the soft link as a recovery mechanism.

In our experiments, we used a fixed window size $w = 100$, and our ideas for further research include the exploration of methods to determine an optimal value for the window size on a per signature basis, perhaps by dynamic adjustment of the window size during the signature generation process. It would also be interesting to know the effect of incorporating phrases into signatures. We have discussed the utility of our approach for the maintenance of links inserted manually, but the method may also be suitable for supporting automatic link discovery, since textual similarity plays a key role in both tasks. For simplicity, we have reported all failed signatures as trace failure. In reality, it is possible to perform further analysis on failed signatures in order to obtain a correct location. We leave these ideas to future work.

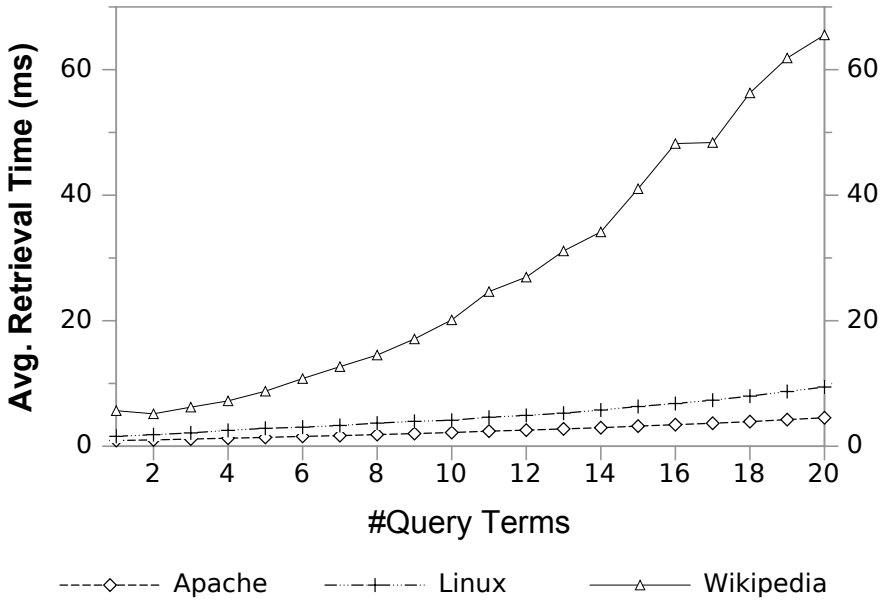


Figure 5.9: Soft link resolution times: Original Collections

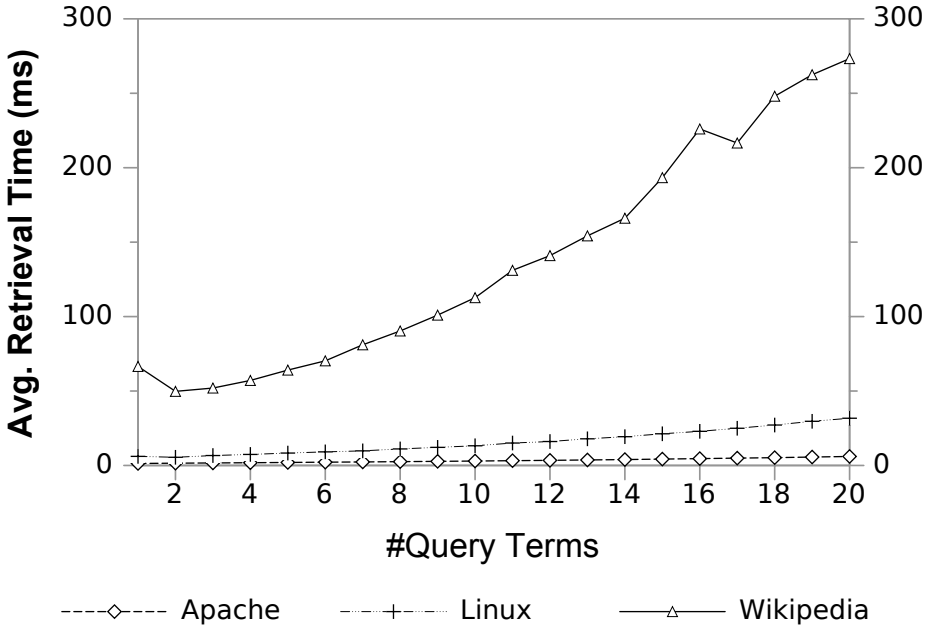


Figure 5.10: Soft link resolution times: Evolved Collections

```
if (loop >= ARRAY_SIZE(ord_data))
    loop = 0;
```

Figure 5.11: A change to the last two lines of Figure 5.1.

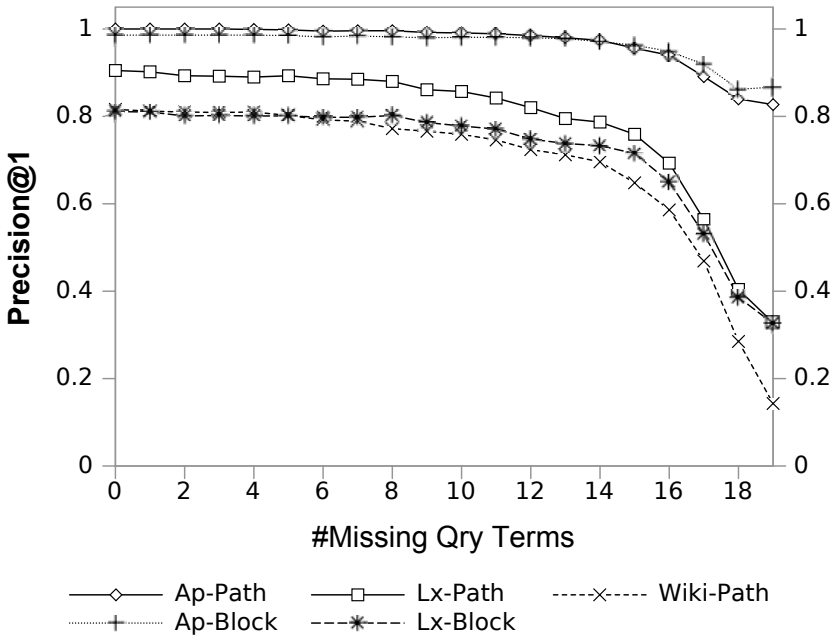


Figure 5.12: Robustness measured by precision@1 as terms are deleted from a signature of size $m = 20$.

Chapter 6

Conclusions

In general, we have explored document similarity as a viable tool for various retrieval-related tasks, especially as it relates to diversity. We have utilized inter-document similarity to provide a means for measuring the quality of subtopic categorization performed by human editorial assessors specifically for diversity-related retrieval tasks in the TREC Web-related experiments. Obviously, more work is required in this area. For example, it would be beneficial if the result of the inter-document similarity measure can be independently compared with respect to the content of the documents by human expert assessors. This may be necessary in order to gain more understanding on the relationship between the judged and the perceived inter-document similarity measures.

We have also provided a framework for evaluating the effectiveness of inter-document similarity measures using the common subtopics between the documents. This idea is predicated upon the cluster hypothesis, especially the diversity-oriented cluster hypothesis, and an assumption that documents sharing the same subtopics should be more similar than those sharing no common subtopics. It would be interesting to investigate the validity of this assumption in greater details. Two datasets from 2009 and 2010 TREC Web Tracks were utilized to investigate this assumption. We believe more datasets should be investigated in order to gain more understanding on the suitability of the cluster hypothesis as well as the assumption that common subtopics reflect inter-document similarity.

On diversity, we provided alternative approaches for uncovering subtopics from text cor-

pora using pseudo-relevance feedback as well as anchor text, anchor out-links, and the target documents the anchor text out-links. These methods did not rely on query and click logs of commercial search engines. Our pseudo-relevance feedback method uncovered terms that are considered important to a query from the top k documents in the retrieved result. The anchor text method utilized anchor text as well as their target documents and the links between them to determine terms that are related to a query. This method even achieved a reasonable result on unicode character representation of the SogouT Chinese corpus in one of our experiments. One would like to see a comparison between the query log generated subtopics and corpus-derived subtopics. An interesting area of future work is the design of more generalized approaches for evaluating the quality of subtopic mining methods. As well, it would be interesting to explore the creation of reusable collections that support diversity-aware retrieval tasks as well as subtopic mining tasks.

We extend our subtopic mining method by utilizing the uncovered subtopics to introduce novelty and diversity into the retrieval result. Mined subtopics are used as query expansion terms. Even though the retrieval is not the particular focus of our method, the expanded queries are used for document retrieval and we obtained reasonable diversity-aware effectiveness result based on both standard and non-standard evaluation measures. Performing a comparative evaluation for our subtopic mining approaches with respect to query and click logs method is challenging. Consequently, it would be difficult to ascertain their quality unless they are compared directly with the query logs method. Research that explores these alternative approaches in conjunction with the query logs mining method on the same document collection will provide a better comparative result for the effectiveness of all the approaches.

This thesis also provides a framework for evaluating diversity–aware ranking functions independent of explicit subtopic categorization. Inter-document similarity was utilized instead of document subtopic information. This is a novel approach for evaluating the effectiveness of diversity-aware ranking functions. An area left for future investigation is the effect of various inter-document similarity measures on the result of the evaluation. In order to measure the similarity of the document at a rank k with all previously seen documents at higher ranks, we implemented a very crude method that concatenates all the tokens. This is one approach. Another approach might combine inter-document similarity scores of documents at higher ranks in an incremental way using a dynamic programming approach. Other approaches might also suffice.

We leave the improvement of this approach as future work.

In this thesis, we have also successfully utilized the similarity between textual data as a fast, effective and robust soft link for tracing specific locations within frequently-edited documents. Our method discovers trace links and recovers broken trace links. This is especially useful for link maintenance in frequently edited documents. The soft links method we provided is generated automatically to maintain traceability links between edit locations in frequently edited documents.

We have implemented our soft links tracing method on three selected datasets. For future work, it would be valuable to implement the soft link method in a real application in order to determine if the method is of actual value to software engineers for tracing and other document linking-related purposes. We have also implemented a fixed window size, we have identified the dynamic determination of optimal window sizes as future work. Various window sizes may also be tried in order to determine an optimal value for the window size of a soft link signature. Rather than the word and character n-gram tokens we implemented, it would be nice to explore the effects of implementing token phrases either by themselves or combined with word and character n-gram tokens. We leave this also for future work.

References

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-390-7. doi: http://doi.acm.org/10.1145/1498759.1498766. 10, 40, 42

John A. Akinyemi and Charles L. A. Clarke. Do subtopic judgments reflect diversity? In *Proceedings of the 3rd International Conference on the Theory of Information Retrieval*, ICTIR '11, pages 309–312, Bertinoro, Italy, 2011. 1, 2, 17, 52

John A. Akinyemi, Charles L. A. Clarke, and Maheedhar Kolla. Towards a collection-based results diversification. In *Proceedings of the 9th RIAO International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 202–205, Paris, France, April 2010. 14, 15, 37, 38, 76

Giuliano Antoniol, Gerardo Canfora, Gerardo Casazza, Andrea De Lucia, and Ettore Merlo. Recovering traceability links between code and documentation. *IEEE Transactions on Software Engineering (TSE)*, 28(10):970–983, 2002. ISSN 0098-5589. doi: {http://dx.doi.org/10.1109/TSE.2002.1041053}. 3, 114

E G. Ashby and Nancy A. Perrin. Toward a unified theory of similarity and recognition. *Psychological Review*, 95:124–150, 1988. 21, 22, 35

Azin Ashkan, Charles L. A. Clarke, Eugene Agichtein, and Qi Guo. Classifying and characterizing query intent. In *ECIR '09: Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval*, pages 578–586, Toulouse, France, 2009. Springer-Verlag. ISBN 978-3-642-00957-0. 39

Javed A. Aslam and Meredith Frost. An information-theoretic measure for document similarity. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 449–450, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3. 1, 4, 26, 51, 53

Ricardo Baeza-Yates, Liliana Calderón-Benavides, and Cristina González-Caro. The intention behind Web queries. In Fabio Crestani, Paolo Ferragina, and Mark Sanderson, editors, *String Processing and Information Retrieval*, volume 4209 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin / Heidelberg, 2006. 39

Ricardo A. Baeza-Yates. Applications of Web query mining. In *European Conference on Information Retrieval Research*, pages 7–22, 2005. 3

Indrajit Bhattacharya and Lise Getoor. Deduplication and group detection using links. In *in Proceedings of the 2004 ACM SIGKDD Workshop on Link Analysis and Group Detection*, 2004. 3

Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the Web. *Computer Networks and ISDN Systems*, 29:1157–1166, September 1997. ISSN 0169-7552. doi: http://dx.doi.org/10.1016/S0169-7552(97)00031-7. URL `http://dx.doi.org/10.1016/S0169-7552(97)00031-7`. 2

Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press, Cambridge, Massachusetts, 2010. 3, 4, 29, 129

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: http://doi.acm.org/10.1145/290941.291025. 37, 39, 76

Ben Carterette and Praveen Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *CIKM '09: Proceeding of the 18th ACM conference on informa-*

tion and knowledge management, pages 1287–1296, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: http://doi.acm.org/10.1145/1645953.1646116. 37, 39, 76

Olivier Chapelle, Donald Metzler, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 621–630, 2009. 10, 40, 42, 66, 67

Chaomei Chen. Structuring and visualising the WWW by generalised similarity analysis. In *Proceedings of the eighth ACM conference on Hypertext*, HYPERTEXT '97, pages 177–186, New York, NY, USA, 1997. ACM. ISBN 0-89791-866-5. doi: http://doi.acm.org/10.1145/267437.267456. URL `http://doi.acm.org/10.1145/267437.267456`. 20

Harr Chen and David R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: http://doi.acm.org/10.1145/1148170.1148245. 39

Shihyen Chen, Bin Ma, and Kaizhong Zhang. The normalized similarity metric and its applications. In *Proceedings of the 2007 IEEE International Conference on Bioinformatics and Biomedicine*, pages 172–180, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3031-1. doi: 10.1109/BIBM.2007.69. URL `http://portal.acm.org/citation.cfm?id=1332472.1332787`. 22, 35

Rudi Cilibrasi and Paul M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005. 31, 53

Charles L. A. Clarke and Egidio L. Terra. Approximating the top m passages in a parallel question answering system. In *13th ACM International Conference on Information and Knowledge Management*, pages 454–462, 2004. 100, 110, 117

Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 358–365, New

York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.384024. 15, 110, 111, 116, 119

Charles L. A. Clarke, Gordon V. Cormack, Thomas R. Lynam, and Egidio L. Terra. Question answering by passage selection. In Tomek Strzalkowski and Sanda Harabagiu, editors, *Advances in Open Domain Question Answering*. Springer, Berlin, Germany, 2006. 14, 111, 116

Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 Web track. In *18th TREC Proceedings*, 2009. 43, 76, 103, 107, 108

Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. Overview of the TREC 2010 Web track. In *19th TREC*, 2010. 107

Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Azin Ashkan. A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the 4th ACM international conference on Web search and data mining (WSDM)*, 2011. 42, 49, 68

Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, July 2008. 5, 10, 37, 40, 42, 76

Jane Cleland-Huang, Raffaella Settimi, Oussama BenKhadra, Eugenia Berezhanskaya, and Selvia Christina. Goal-centric traceability for managing non-functional requirements. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, pages 362–371, New York, NY, USA, 2005. ACM. ISBN 1-59593-963-2. doi: {http://doi.acm.org.proxy.lib.uwaterloo.ca/10.1145/1062455.1062525}. 113

Jane Cleland-Huang, Jane H. Hayes, and Jean M. Domel. Model-based traceability. *In ICSE Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)*, pages 6–10, May 2009. 3

Paul Clough, Mark Sanderson, Murad Abouammoh, Sergio Navarro, and Monica Paramita. Multiple approaches to analysing query diversity. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 734–735, Boston, MA, USA, 2009. ACM. ISBN 978-1-60558-483-6. 38, 39

James W. Cooper, Anni R. Coden, and Eric W. Brown. Detecting similar documents using salient terms. In *Proceedings of the eleventh international conference on Information and knowledge management*, CIKM '02, pages 245–251, New York, NY, USA, 2002. ACM. ISBN 1-58113-492-4. 1, 2

W. B. Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 2009. ISBN 0136072240, 9780136072249. 4, 29

Steve Cronen-Townsend and W. B. Croft. Quantifying query ambiguity. In *Proceedings of the second international conference on Human Language Technology Research*, pages 104–109, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. 38

Van Dang and W. B. Croft. Query reformulation using anchor text. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 41–50, 2010. ISBN 978-1-60558-889-6. 38, 90

Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1251254.1251264`. 2

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science (JASIS)*, 41(6):391–407, 1990. 21

Fernando Diaz. Regularizing query-based retrieval scores. *Information Retrieval*, 10(6):531–562, 2007. 34

Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, July 1945. 30, 53

Ekwa Duala-Ekoko and Martin P. Robillard. Tracking code clones in evolving software. In *International Conference on Software Engineering (ICSE)*, 2007. 3, 114

N. Eiron and K. S. McCurley. Analysis of anchor text for Web search. In *26th ACM SIGIR*, pages 459–460, 2003. 90

Tamer Elsayed, Jimmy Lin, and Douglas W. Oard. Pairwise document similarity in large collections with MapReduce. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 265–268, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://portal.acm.org/citation.cfm?id=1557690.1557767`. 3

Michelle Girvan and M. E. Newman. Community structure in social and biological networks. pages 7821–7826, June 2002. 80, 94

Michael W. Godfrey and Lijie Zou. Using origin analysis to detect merging and splitting of source code entities. *IEEE Transactions on Software Engineering (TSE)*, 31(2):166–181, 2005. ISSN 0098-5589. doi: http://dx.doi.org/10.1109/TSE.2005.28. 113

Gene Golovchinsky. What the query told the link: the integration of hypertext and information retrieval. In *HYPERTEXT '97: Proceedings of the eighth ACM conference on Hypertext*, pages 67–74, New York, NY, USA, 1997. ACM. ISBN 0-89791-866-5. doi: {http://doi.acm.org/10.1145/267437.267445}. 115

Gene Golovchinsky and Mark Chignell. Queries-R-Links: graphical markup for text navigation. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 454–460, New York, NY, USA, 1993. ACM. ISBN 0-89791-575-5. doi: {http://doi.acm.org/10.1145/169059.169372}. 115

O. C. Z. Gotel and C. W. Finkelstein. An analysis of the requirements traceability problem. In *Requirements Engineering, 1994., Proceedings of the First International Conference on*, pages 94 –101, Apr 1994. doi: {10.1109/ICRE.1994.292398}. 112

Edward Grefenstette. Analysing document similarity measures. Master's thesis, University of Oxford, September 2009. URL `http://{Web}.comlab.ox.ac.uk/people/edward.grefenstette/MScThesis.pdf`. 20

Maen Hammad, Michael L. Collard, and Jonathan I. Maletic. Automatically identifying changes that impact code-to-design traceability. In *International Conference on Program Comprehension (ICPC)*, pages 20–29, 2009. 113

S. P. Harter and C. A. Hert. Evaluation of information retrieval systems: Approaches, issues, and methods. *Annual Review of Information Science and Technology*, 32:3–94, 1997. URL `http://www.eric.ed.gov/ERICWebPortal/detail?accno=EJ565471`. 3

Jane H. Hayes, Alex Dekhtyar, Senthil K. Sundaram, Elizabeth A. Holbrook, Sravanthi Vadlamudi, and Alain April. REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery. *ISSE*, 3(3):193–202, 2007. 3, 114

Jane Huffman Hayes, Alex Dekhtyar, and James Osborne. Improving requirements tracing via information retrieval. In *RE '03: Proceedings of the 11th IEEE International Conference on Requirements Engineering*, page 138, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1980-6. 113, 114

J. He, E. Meij, and M. de Rijke. Result diversification based on query-specific cluster ranking. *JASIST*, 62(3):550–571, 2011. 39, 107

Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *19th SIGIR*, pages 76–84, 1996. 43

Nevin Heintze. Scalable document fingerprinting. In *In Proceeding of the USENIX Workshop on electronic commerce*, 1996. 2

Mauro Rojas Herrera, Edleno S. de Moura, Marco Cristo, Thomaz P. Silva, and Altigran S. da Silva. Exploring features for the automatic identification of user goals in Web search. *Information Process Management*, 46(2):131–142, 2010. ISSN 0306-4573. 39

D. Hiemstra and C. Hauff. MIREX: MapReduce information retrieval experiments. Technical Report TR-CTIT-10-15, Centre for Telematics and Information Technology University of Twente, Enschede, April 2010. 93, 103

Jian Hu, Gang Wang, Fred Lochovsky, Jian-Tao Sun, and Zheng Chen. Understanding user's query intent with Wikipedia. In *WWW '09: Proceedings of the 18th international conference*

on World wide Web, pages 471–480, Madrid, Spain, 2009. ACM. ISBN 978-1-60558-487-4. 38

Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the user intent of Web search engine queries. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1149–1150, Banff, Alberta, Canada, 2007. ACM. ISBN 978-1-59593-654-7. 39

Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of Web queries. *Information Process Management*, 44(3): 1251–1266, 2008. ISSN 0306-4573. 39

N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217 – 240, 1971. 50

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/582415.582418. 40

Hsin-Yi Jiang, T. N. Nguyen, Chen Ing-Xiang, H. Jaygarl, and C. K. Chang. Incremental latent semantic indexing for automatic traceability link evolution management. In *Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 59–68, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-1-4244-2187-9. 112, 113

Karen Jones, Stephen Robertson, Djoerd Hiemstra, and Hugo Zaragoza. Language modeling and relevance. In W. B. Croft and John Lafferty, editors, *Language Modeling for Information Retrieval*, volume 13 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers, 2003. 28

Sunghun Kim, Kai Pan, and E. J. Whitehead, Jr. When functions change their names: Automatic detection of origin relationships. In *WCRE '05: Proceedings of the 12th Working Conference on Reverse Engineering*, pages 143–152, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2474-5. doi: http://dx.doi.org/10.1109/WCRE.2005.33. 113

R. Kraft and J. Zien. Mining anchor text for query refinement. In *13th WWW Conference*, pages 666–674, 2004. 90

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951. 23, 29

Oren Kurland. *Inter-document similarities, language models, and ad hoc information retrieval*. PhD thesis, Cornell University, 2006. 2, 9, 28, 29, 34

Oren Kurland and Lillian Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the international ACM SIGIR conference on research and development in information retrieval*, pages 194–201, 2004. 1, 2, 9, 29

Thomas K. Landauer and Susan T. Dumais. A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997. 21

Victor Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.383972. URL `http://doi.acm.org/10.1145/383952.383972`. 28

Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul M. B. Vitányi. The similarity metric. *IEEE Transactions on Information Theory*, pages 3250–3264, 2004. 22

Dekang Lin. An information-theoretic definition of similarity. In *International Conference on Machine Learning*, pages 296–304, 1998. 1, 25, 53

Jimmy Lin. Brute force and indexed approaches to pairwise document similarity comparisons with MapReduce. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 155–162, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1571970. URL `http://doi.acm.org/10.1145/1571941.1571970`. 1, 2

P. Mader, O. Gotel, and I. Philippow. Enabling automated traceability maintenance by recognizing development activities applied to models. In *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 49–58, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-1-4244-2187-9. doi: http://dx.doi.org/10.1109/ASE.2008.15. 113

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715. 3

Paul McNamee and James Mayfield. Character n-gram tokenization for European language text retrieval. *Information Retrieval*, 7(1-2):73–97, 2004. URL `http://www.springerlink.com/content/lr172kr538374510/`. 123

Paul McNamee, Charles Nicholas, and James Mayfield. Don't have a stemmer?: Be un+concern+ed. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 813–814, Singapore, 2008. URL `http://doi.acm.org/10.1145/1390334.1390518`. 123

Lior Meister, Oren Kurland, and Inna G. Kalmanovich. Re-ranking search results using an additional retrieved list. *Information Retrieval*, 14(4):413–437, 2010. 1, 2, 9, 29

Timothy B. Mundie and Frederick J. Hallsworth. Requirements analysis using SuperTrace PC. Houston, Texas, USA, 1995. American Society of Mechanical Engineers (ASME). 112

Leonardo G. P. Murta, Andre van der Hoek, and Claudia M. L. Werner. ArchTrace: Policy-based support for managing evolving architecture-to-implementation traceability links. In *International Conference on Automated Software Engineering (ASE)*, pages 135–144, 2006. 113

Christopher Olston and Ed H. Chi. ScentTrails: Integrating browsing and searching on the Web. *ACM Transactions on Computer-Human Interaction*, 10:177–197, September 2003. 115

Jay M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and*

development in information retrieval, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: http://doi.acm.org/10.1145/290941.291008. URL `http://doi.acm.org/10.1145/290941.291008`. 2, 28

John Prager. Open-domain question-answering. *Foundations and Trends in Information Retrieval*, 1(2):91–231, 2006. 116

R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17 –30, jan/feb 1989. ISSN 0018-9472. doi: 10.1109/21.24528. 24, 35

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *ICML '08: Proceedings of the 25th international ACM conference on Machine learning*, pages 784–791, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390255. 40

Filip Radlinski, Martin Szummer, and Nick Craswell. Inferring query intent from reformulations and clicks. In *WWW '10: Proceedings of the 19th international conference on World wide Web*, pages 1171–1172, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: http://doi.acm.org/10.1145/1772690.1772859. 10, 37, 39, 76, 90, 103, 107

Davood Rafiei, Krishna Bharat, and Anand Shukla. Diversifying Web search results. In *WWW '10: Proceedings of the 19th international conference on World wide Web*, pages 781–790, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: http://doi.acm.org/10.1145/1772690.1772770. 38, 39, 40

Balasubramaniam Ramesh. Factors influencing requirements traceability practice. *Communications of the ACM*, 41(12):37–44, 1998. ISSN 0001-0782. doi: {http://doi.acm.org/10.1145/290133.290147}. 112

R. Real and J. M. Vargas. The probabilistic basis of Jaccard's index of similarity. *Systematic Biology*, 45(3):380–385, 1996. 30, 53

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *CIKM '04: Proceedings of the thirteenth ACM international conference*

on Information and knowledge management, pages 42–49, New York, NY, USA, 2004. ACM. ISBN 1-58113-874-1. doi: http://doi.acm.org/10.1145/1031171.1031181. 24, 33

Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.*, 18(2):95–145, 2003. ISSN 0269-8889. doi: http://dx.doi.org/10.1017/S0269888903000638. 77

Tetsuya Sakai and Ruihua Song. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, pages 1043–1052, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: http://doi.acm.org/10.1145/2009916.2010055. URL `http://doi.acm.org/10.1145/2009916.2010055`. 10, 41

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communication of the ACM*, 18:613–620, November 1975. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/361219.361220. URL `http://doi.acm.org/10.1145/361219.361220`. 21, 26, 53

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840. 9

C. Santamaria, J. Gonzalo, and J. Artiles. Wikipedia as sense inventory to improve diversity in Web search results. In *Proceedings of ACL 2010*, Uppsala, Sweden, 2010. Association for Computational Linguistics. 38

Simone Santini and Ramesh Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:871–883, 1999. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/34.790428. 22, 35

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for Web search result diversification. In *WWW'10*, pages 881–890, 2010a. 38, 39

Rodrygo L. T. Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. Explicit search result diversification through sub-queries. In *European Conference on Information Retrieval Research (ECIR)*, pages 87–99, 2010b. 37, 76

Rao Shen, Naga S. Vemuri, Weiguo Fan, Ricardo da S. Torres, and Edward A. Fox. Exploring digital libraries: Integrating browsing, searching, and visualization. In *6th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 1–10, 2006. 116

Mark D. Smucker and James Allan. Find-similar: similarity browsing as a search tool. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 461–468, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: http://doi.acm.org/10.1145/1148170.1148250. URL `http://doi.acm.org/10.1145/1148170.1148250`. 1, 2

Mark D. Smucker and James Allan. A new measure of the cluster hypothesis. In *2nd International Conference on the Theory of Information Retrieval*, pages 281–288, 2009. 4, 43, 50, 51

R. Song, M. Zhang, T. Sakai, M. P. Kato, Y. Liu, M. Sugimoto, Q. Wang, and N. Orii. Overview of the NTCIR-9 INTENT Task, NTCIR-9 Proceedings. Tokyo, Japan., December 2011. NII. 42, 77, 95

Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. Identification of ambiguous queries in Web search. *Information Process Management*, 45(2):216–229, 2009. ISSN 0306-4573. 39, 76, 103

Benno Stein. Principles of hash-based text retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 527–534, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: http://doi.acm.org/10.1145/1277741.1277832. URL `http://doi.acm.org/10.1145/1277741.1277832`. 2

Benno Stein and Martin Potthast. Applying hash-based indexing in text-based information retrieval. 2007. 2

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 41–47, 2003. 117

Anastasios Tombros and C. J. van Rijsbergen. Query-sensitive similarity measures for the calculation of interdocument relationships. In *CIKM*, pages 17–24, 2001. 50

A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977. 1, 20, 21, 22, 25, 35, 52

Ozlem Uzuner, Randall Davis, and Boris Katz. Using empirical methods for evaluating expression and content similarity. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4 - Volume 4*, HICSS '04, pages 40104.1–, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2056-1. URL `http://portal.acm.org/citation.cfm?id=962752.962960`. 1, 3

C. J. van Rijsbergen. *Information Retrieval* (2nd edition). Butterworths, London, 1979. 4, 24, 33, 43

Ellen M. Voorhees. The cluster hypothesis revisited. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '85, pages 188–196, New York, NY, USA, 1985. ACM. ISBN 0-89791-159-8. 4, 43, 50

Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, SIGIR '09, pages 115–122, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. 40

Xuanhui Wang, Bin Tan, Azadeh Shakery, and Chengxiang Zhai. Beyond hyperlinks: Organizing information footprints in search logs to support effective browsing. In *18th ACM Conference on Information and Knowledge Management*, pages 1237–1246, 2009. 115

Yisong Yue and Thorsten Joachims. Predicting diverse subsets using structural SVMs. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1224–1231, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390310. 40

ChengXiang Zhai and John Lafferty. A risk minimization framework for information retrieval. *Information Process Management*, 42:31–55, January 2006. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2004.11.003. URL `http://dx.doi.org/10.1016/j.ipm.2004.11.003`. 40

ChengXiang Zhai, William W. Cohen, and John D. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the international ACM SIGIR conference on research and development in information retrieval*, pages 10–17, 2003. 39